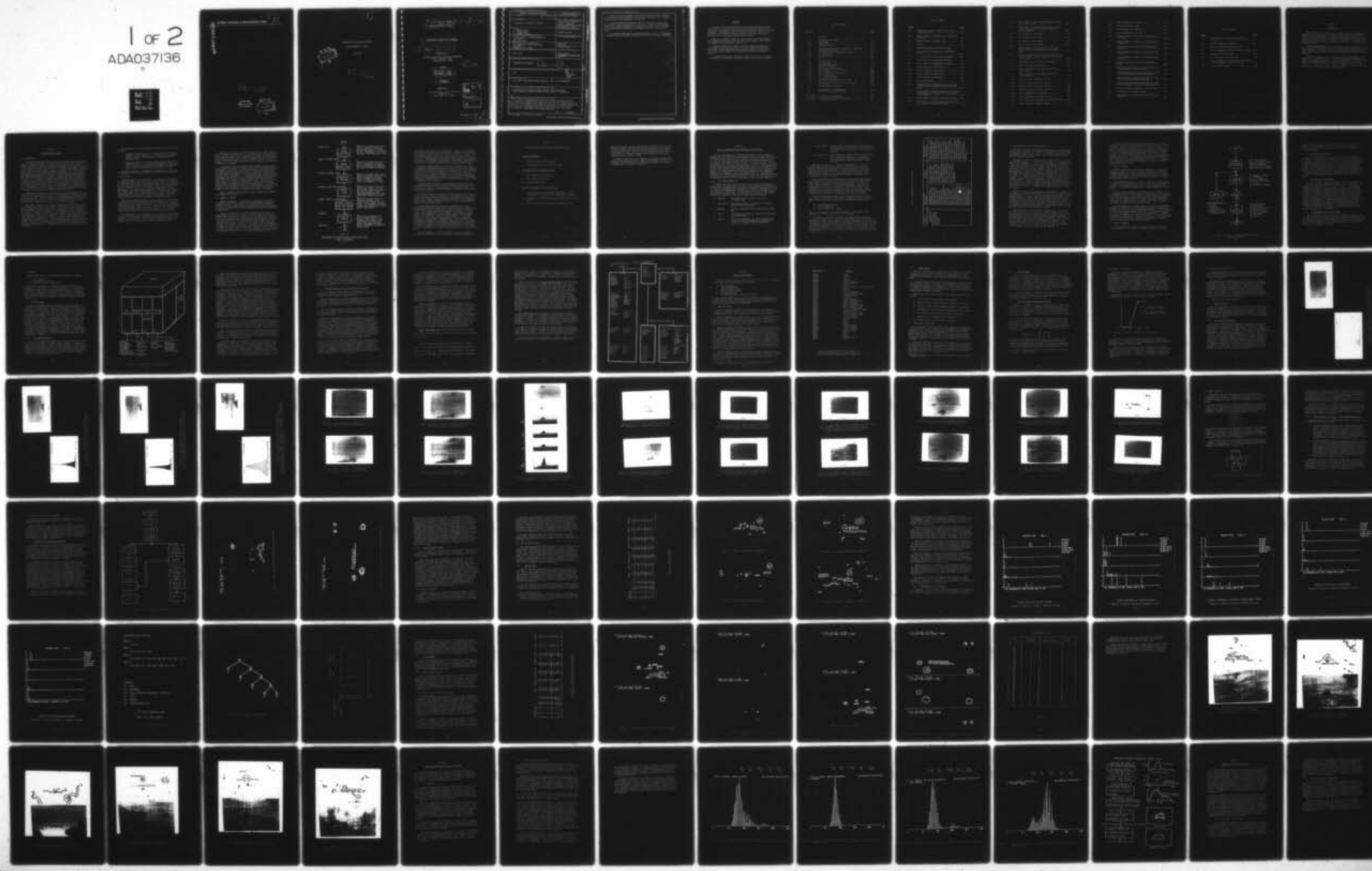
AD-A037 136 PATTERN ANALYSIS AND RECOGNITION CORP ROME N Y F/G 17/5
PROCESSING OF FLIR DATA ON DICIFER.(U)
JUL 76 D J LUCEY, G E FORSEN, M J ZORACKI DAAG53-75-C-0277
UNCLASSIFIED PAR-76-26 NL

1 OF 2
ADA037136



ADA037136

PATTERN ANALYSIS & RECOGNITION CORP.

DISTRIBUTION STATEMENT A
Approved for public release;
Distribution Unlimited

ON THE MALL
ROME, N. Y. 13440
Tel. 315-336-8400

(1)

PROCESSING OF FLIR DATA ON DICIFER

Contract #DAAG53-75-C-0277

July 1976

(9) FINAL TECHNICAL REPORT. Jul 75- Jun 76,

(15) Contract #DAAG53-75-C-0277

(6) PROCESSING OF FLIR DATA ON DICIFER.

(14) PAR-76-26

SUBMITTED BY:

PATTERN ANALYSIS AND RECOGNITION CORPORATION
228 W. Dominick Street
Rome, New York 13440

(12) 108p.

AUTHORS

(10) Mr. Dennis J. Lucey
Mr. George E. Forsen
Dr. Michael J. Zoracki

(11) JULY [redacted]76

ACCESSION for		
NTIS	White Section	☑
DDC	Buff Section	☐
UNANNOUNCED		☐
JUSTIFICATION		
BY		
DISTRIBUTION/AVAILABILITY CODES		
Dist.	AVAIL. and/or SPECIAL	
A		

PRESENTED TO:

U.S. Army Night Vision Laboratory
DRSEL-NV-VI
Fort Belvoir, VA 22060

PAR Report No. 76-26

dm

390 101

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER	
2. GOVT ACCESSION NO.	
3. RECIPIENT'S CATALOG NUMBER	
4. TITLE (and Subtitle)	Processing of FLIR Data on DICIFER
5. TYPE OF REPORT & PERIOD COVERED	Final Technical Report July 75 - June 76
6. PERFORMING ORG. REPORT NUMBER	PAR Report No. 76-26
7. AUTHOR(s)	Mr. Dennis Lucey Mr. George Forsen Dr. Michael Zoracki
8. CONTRACT OR GRANT NUMBER(s)	DAAG53-75-C-0277
9. PERFORMING ORGANIZATION NAME AND ADDRESS	Pattern Analysis & Recognition Corp. On The Mall Rome, NY 13440
10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS	
11. CONTROLLING OFFICE NAME AND ADDRESS	U.S. Army Night Vision Laboratory DRSEL-NV-VI Ft Belvoir, VA 22060
12. REPORT DATE	July 1976
13. NUMBER OF PAGES	
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)	Same as 11
15. SECURITY CLASS. (of this report)	Unclassified
15a. DECLASSIFICATION/DOWNGRADING SCHEDULE	N/A

16. DISTRIBUTION STATEMENT (of this Report)

Distribution Unlimited

DISTRIBUTION STATEMENT A
Approved for public release;
Distribution Unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

Same

18. SUPPLEMENTARY NOTES

Night Vision Laboratory Project Engineer: Mr. G. David Singer

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Forward-looking Infrared Imagery, Digital Image Processing, Image Processing, Automatic Target Cueing, Pattern Recognition

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This effort has investigated the extent to which automatic tactical target cueing on selected FLIR (forward-looking infrared) imagery can be accomplished using digital image processing and automatic pattern recognition techniques. The DICIFER (Digital Interactive Complex for Image Feature Extraction and Recognition) system was utilized to analyze FLIR image data for the U.S. Army Night Vision Laboratory (NVL).

(CONT'D)

DICIFER is a general purpose R&D tool developed by Pattern Analysis and Recognition (PAR) Corporation for Rome Air Development Center (RADC), Intelligence and Reconnaissance Division (IR).

A digital process for automatic detection of tactical targets in FLIR images was designed and tested. The total process included noise removal and contrast enhancement preprocessing; gradient edge detection and boundary chain encoding; and the design of Boolean classification logic.

The classification logic was applied against a test set of 34 images. The probability of detection achieved was 88% (38 of 43 targets detected).

ABSTRACT

This effort has investigated the extent to which automatic tactical target cueing on selected FLIR (forward-looking infrared) imagery can be accomplished using digital image processing and automatic pattern recognition techniques. The DICIFER (Digital Interactive Complex for Image Feature Extraction and Recognition) system was utilized to analyze FLIR image data for the U.S. Army Night Vision Laboratory (NVL).

DICIFER is a general purpose R&D tool developed by Pattern Analysis and Recognition (PAR) Corporation for Rome Air Development Center (RADC), Intelligence and Reconnaissance Division (IR).

A digital process for automatic detection of tactical targets in FLIR images was designed and tested. The total process included noise removal and contrast enhancement preprocessing; gradient edge detection and boundary chain encoding; and the design of Boolean classification logic.

The classification logic was applied against a test set of 34 images. The probability of detection achieved was 88% (38 of 43 targets detected).

TABLE OF CONTENTS

Section		Page
1.	Objective	1-1
2.	Introduction and Summary	2-1
2.1.	Background	2-1
2.2.	The Data Set	2-2
2.3.	Summary of Results	2-3
3.	DICIFER System Background, Philosophy, & Functions	3-1
3.1.	IFES/SCORE Development	3-1
3.2.	The General Pattern Recognition Problem	3-4
3.3.	System Philosophy	3-5
3.4.	Facility Hardware	3-5
3.5.	Software	3-8
4.	Technical Discussions	4-1
4.1.	Data Selection	4-1
4.2.	Image Noise Attenuation	4-3
4.3.	Contrast Enhancement	4-5
4.4.	Comments on Preprocessing	4-6
4.5.	Introduction to Target Detection	4-6
4.6.	Edge Detection	4-20
4.7.	Approach for FLIR Target Detection	4-22
4.8.	Feature Extraction	4-26
4.9.	Object Classification	4-27
4.10.	Classification Results	4-40
4.11.	Nonclassified Targets and False Alarms	4-40
5.	Further Considerations for Target Detection	5-1
5.1.	Prescreening	5-1
5.2.	Automatic Subimage Analysis	5-2
6.	Conclusions and Recommendations	6-1
6.1.	Topics for Future Investigations	6-1
References		R-1
Appendix A	Boundary Tracing Algorithm	A-1
Appendix B	Sub-Image Single Threshold Calculation	B-1
Appendix C	FLIR Data Assessment	C-1

LIST OF FIGURES

Figure		Page
2-1	Processing for Single Frame Target Detection (ANVL FLIR Data)	2-4
3-1	Typical Process for Solving Pattern Recognition Problems	3-6
3-2	Building Blocks of Software System	3-9
3-3	DICIFER	3-14
4-1	Transfer Function Used for Range Change	4-5
4-2	Frame K00010 as Displayed on the DICIFER TV Monitor	4-7
4-3	Frame K100D0 and Corresponding Histogram	4-8
4-4	Frame K10SP0 and Corresponding Histogram	4-9
4-5	Frame K10SP1 and Corresponding Histogram	4-10
4-6	Central Portion of Frame K00010	4-11
4-7	Central Portion of Frame K100D0	4-11
4-8	Central Portion of Frame K10SP0	4-12
4-9	Central Portion of Frame K10SP1	4-12
4-10	Grey Level Profiles	4-13
4-11	Display Showing Boundaries Extracted from Frame K00010 Prior to the Application of the Cueing Logic	4-14
4-12	Display Showing Boundary Remaining after Filtering with Boolean Logic Designed to Detect Tactical Target Shapes	4-14
4-13	Frame C00004 as Displayed on DICIFER TV Monitor	4-15
4-14	Frame C40D00. Results of Processing Frame C00004 for Removal of Salt-and-Pepper Noise	4-15

4-15 Frame C4SP00. Results of Applying Spatial Filtering to Frame C40D00. 4-16

4-16 Frame C4SP01. Results of Applying Dynamic Range Change to Frame C4SP00 in Order to Achieve Contract Enhancement. 4-16

4-17 Left Portion of Frame C00004 4-17

4-18 Left Portion of Frame C40D00. 4-17

4-19 Left Portion of Frame C4SP00. 4-18

4-20 Left Portion of Frame C4SP01. 4-18

4-21 Display Showing Boundaries Extracted from Frame C00004 Prior to the Application of the Cueing Logic. 4-19

4-22 Display Showing Boundaries Remaining after Filtering with Boolean Logic Designed to Detect Tactical Target Shapes. 4-19

4-23 Neighborhoods Used for Gradient Calculation . . . 4-20

4-24 Target Detection Processing Flow. 4-23

4-25 Results of Applying "Area Edge Detection" To Frame K00010. 4-24

4-26 Results of Applying "Area Edge Detection" To Frame C00004. 4-25

4-27 Display of Chain Vectors for Frame K4. 4-29

4-28 Display of Chain Vectors for Frame L4. 4-29

4-29 Display of Chain Vectors for Frame L5. 4-30

4-30 Display of Chain Vectors for Frame L6. 4-30

4-31 Sample Histograms on Area Feature. 4-32

4-32 Sample Histograms on Perimeter Feature 4-33

4-33 Sample Histograms on Perimeter Squared/Area Feature 4-34

4-34 Sample Histograms of Width Feature. 4-35

4-35	Sample Histograms on Height Feature	4-36
4-36	Structure of Boolean Logic.	4-38
4-37	Decision Images for Image K10.	4-42
4-38	Some Decision Images for Image C4	4-45
4-39	Decision Images for Frame K6 Illustrating Missed Detection.	4-48
4-40	Decision Images for Frame H8 Illustrating Missed Detection	4-49
4-41	Decision Images for Frame E7 Illustrating Missed Detection	4-50
4-42	Decision Images for Frame E6 Illustrating Missed Detection	4-51
4-43	Decision Images for Frame E5 Illustrating Missed Detection	4-52
4-44	Decision Images for Frame B5 Illustrating False Alarm	4-53
5-1	Histogram and Statistical Calculations for a Sub-image Containing a Tactical Target.	5-4
5-2	Histogram and Statistical Calculations for a Sub-image Containing Only "Smooth" Background	5-5
5-3	Histogram and Statistical Calculations for a Sub-image Containing a "Noisy" Background.	5-6
5-4	Histogram and Statistical Calculations for a Sub-image Containing a Portion of a Non-Target Object	5-7
5-5	Processing Sequence Proposed for Target Cueing. .	5-8
B-1	Decomposition of Bimodal Histogram.	B-4
B-2	Flow Diagram of Sub-Image Single Threshold Calculation	B-6

LIST OF TABLES

Table		Page
2-1	Characteristics of Algorithms Proposed for Cueing	2-6
3-1	System Capabilities of DICIFER.	3-3
4-1	Images and Targets Making Up Data Set	4-2
4-2	Header Information for Chain Vector File C4CH00	4-28
4-3	Description of Boolean Logic.	4-37
4-4	Confusion Matrix for Design Logic	4-39
4-5	Header Information for Chain Vector File C4CH00 with Classification Label (Column ID). . . .	4-41
4-6	Classification Results.	4-46

SECTION 1

OBJECTIVE

The objective of this effort was to determine the extent to which automatic tactical target cueing on selected forward-looking infrared (FLIR) imagery could be accomplished using digital image processing and automatic pattern recognition techniques. During the course of this effort the DICIFER (Digital Interactive Complex for Image Feature Extraction and Recognition) system was utilized to analyze FLIR image data for the U.S. Army Night Vision Laboratory (NVL).

The digitized FLIR images used for this investigation were supplied by NVL. Ground data was supplied by NVL in the form of photointerpreter analyses which specified the number and types of tactical targets contained in each image. Target types included tank, armored personnel carrier (APC) and 2-1/2 ton truck.

This effort represents one of four parallel efforts sponsored by NVL to assist their evaluation of the use of digital image processing techniques to enhance FLIR operator performance. This report describes the study made under Contract No. DAAG53-75-C-0277 and covers the period from July 1975 through June 1976. This period includes a six-month extension required to cover down time incurred due to failure of the disc memory sync track on the DICIFER video display and the time required for its repair.

SECTION 2

INTRODUCTION AND SUMMARY

2.1. BACKGROUND

The motivation for providing cueing aids to the already busy helicopter pilot is easily appreciated. A short example in geometry will provide one reason. Consider a friendly helicopter traveling over hostile territory from point A to point B. The safest ground fire avoidance flight mode is said to be the highest V/H ratio, which is just the worst case for keeping a target in view in the direction of flight. Assume, for example, that the horizon is in view at the top of the display and the depression angle is 7-1/2°. Under these conditions the bottom half of the screen shows ground coverage about four times the altitude of the platform. Targets appearing at the bottom of the screen will be reached the soonest and will be the first to leave the display. If the helicopter is moving in the direction the FLIR is pointed, at, say, 150 mph at an AGL altitude of 200 feet, it will take less than two seconds for a target to move from the center of the display off the bottom end! Lower AGL altitudes are worse. It seems unlikely that the human operator could respond to a target either of opportunity or of necessity in that time, especially when it is not expected or when there are priority judgments to be made. Thus early, i.e., distant, detection is needed.

Target search in the bob-up mode requires the helicopter to rise up from behind its tree cover and for as brief a time as possible survey the neighboring area. The data supplied by NVL for this application was obtained during tests of this type. The scenes generally contained unobscured targets in an open field with some trees and scrub. Occasionally targets were partially hidden either by objects in the scene or by noise in the image.

Detection of targets in FLIR imagery depends very heavily on the interpretation abilities of the human analyst. Thus, when we speak of probabilities of detection, we must deal not only with a given sensor's capacity to record infrared energy, but also with the probability that the interpreter will detect a target in the image data. As already mentioned, the difficulty of the operator's task is greatly increased when he is operating under severe time constraints in a hostile environment. Digital image processing techniques were investigated on this effort to evaluate their potential to increase detection probabilities. The techniques used were available on DICIFER (Digital Interactive Complex for Image Feature Extraction and Recognition, located at Rome Air Development Center) to analyze FLIR data for U.S. Army Night Vision Laboratory. The DICIFER software system was supplied to the Air Force by PAR.

Two primary areas for digital image processing during this effort have been:

- o Automatic Target Cueing, i.e., to direct the human operator's attention to those portions of the display which are likely to contain a target of interest having a high probability of detection), and

- o Image Enhancement, i.e., to improve the apparent target-to-background contrast ratios, attenuate image noise, and improve overall image quality so that both contextual information and target are more easily perceived by the human operator.

Specifics regarding the algorithms utilized and results achieved are presented in the technical discussions in Section 4 of this report.

2.2. THE DATA SET

FLIR image data was supplied by NVL. The FLIR system is sensitive to thermal radiation. Video patterns created by differences in effective radiation temperatures were recorded on video tape during maneuvers that were intended to test operator performance in a bob-up survey made. The tests were made two hours past sundown. Most of the targets that showed detectable contrast were or had been moving under their own engine power. Thus their engine compartments and exhaust areas radiated more thermal energy than did their background. Trees and shrubs still appeared brighter, as did certain tracks on the ground. The target characteristics which appeared to have potential for automatic processing included large size, high contrast, and distinct edges.

The FLIR data was generated for a high-resolution display and was converted by NVL to images containing about 800,000 pixels. This is a large amount of data to process in "real time". Thus image enhancement and target detection algorithms were kept simple so they could be implemented without great expense per unit. However, the data tested on this project covered the entire range of image quality. Excessive high contrast interference noise patterns occurred in much of the data.

A working test set of image data was chosen from the data supplied by NVL. The images used and reported herein were selected to be representative of the variety of scenes and image qualities provided. The data set included 38 images, which contained a total of 47 tactical targets (i.e., 21 tanks, 18 APC's, and eight 2-1/2 ton trucks).

The data set used for this work included the short range scenes in which the targets are generally in the foreground and appear relatively large. Specific image frames are listed in Section 4. Although there was a slight variance in scale among the images selected, the differences were not significant enough to cause difficulties in handling scale changes. For this reason we were able to use a single set of parameters to process all 38 images.

It is important to note, however, that the detection logic implemented during this effort was based on shape and size parameters derived from edges extracted from scene objects (targets). These measurements (in terms of picture elements) are directly affected by image scale. For this reason processing of the smaller scale images also would have required different parameters. It is conceivable that algorithms can be developed which would make use of informations derived directly from the FLIR control and ancillary sources, including altitude, depression angle, and field of view to determine ground scale. Such an algorithm would provide for automatic adjustments to critical parameters and should enhance detection and classification accuracies. The hypotheses mentioned in this paragraph could not be tested within the scope of this program. These concepts remain as topics for future investigations which would require additional algorithm and software development.

Details regarding characteristics of the image data are presented in Section 4 of this report.

2.3. SUMMARY OF RESULTS

During this effort, we investigated the application of existing techniques on DICIFER for the purposes of determining an approach to automatic tactical target cueing. In addition, it was necessary to perform some image preprocessing by applying digital spatial filtering to reduce the effects of noise and to enhance the target-to-background contrast ratios.

Figure 2-1 reviews the processing sequence used to enhance the image quality and to detect targets. The noise filtering shown was required only to remove the effects of noise peculiar to the data used and is not a necessity for FLIR post-processing, in general. Thus, the choice of filter (if needed) would depend on the peculiarities of the particular FLIR design. On the other hand, the detection logic and features used are scene-dependent. They require a priori knowledge of the types of targets to be detected. Additional information that can be derived directly from the FLIR control and ancillary sources includes altitude, depression angle, and FOV. Any combination that gives ground scale greatly enhances detection and classification accuracies. The

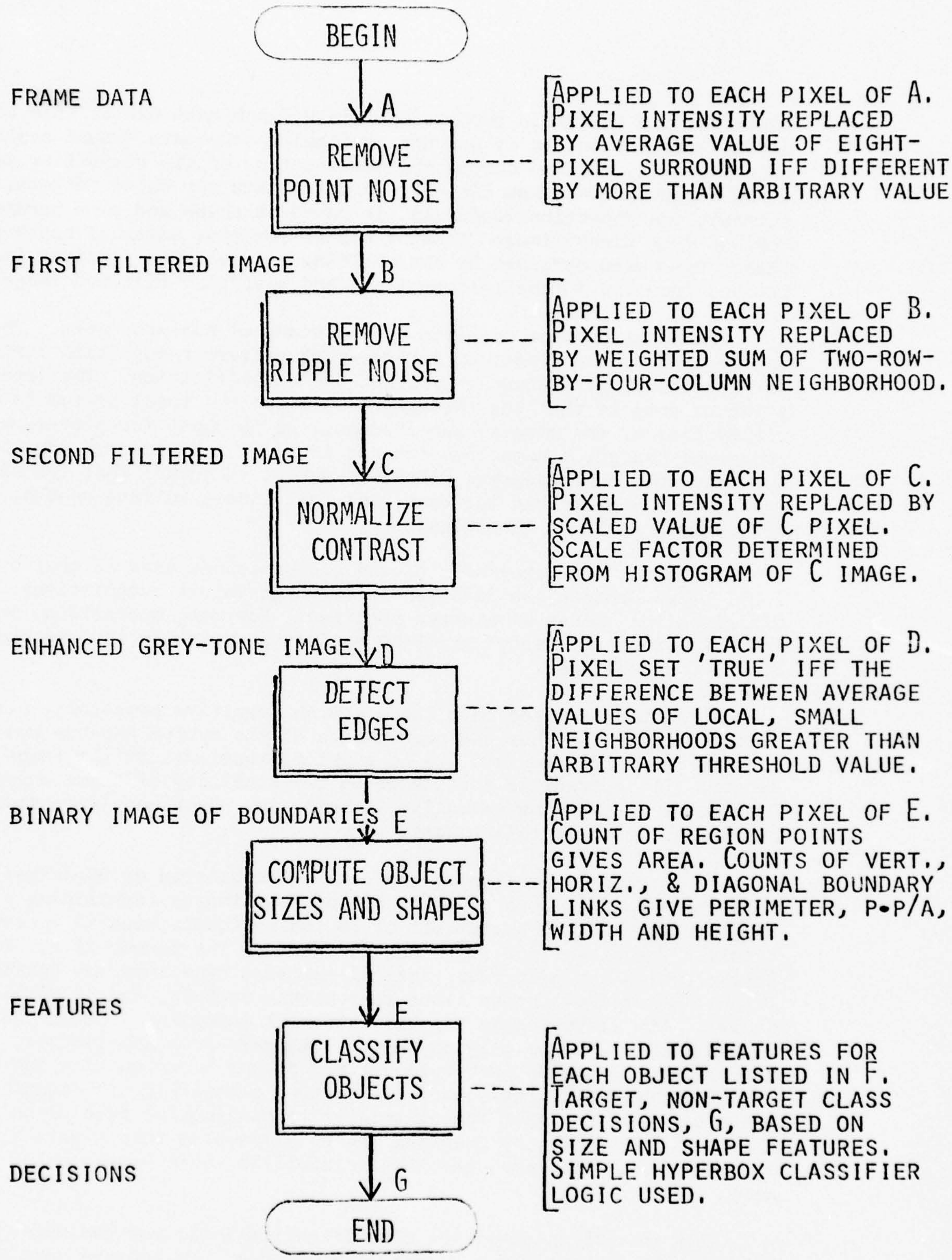


FIGURE 2-1

PROCESSING FOR SINGLE FRAME TARGET DETECTION
(ANVL FLIR DATA)

logic used in the tests reported herein did not make use of this additional information, as it was not available. Adequate ground scale was experimentally derived during the construction of the classifier logic using measurements taken directly from the data for known targets. In summary, the detection logic used is based on shape and size parameters derived from binary images showing higher-contrast edges of scene objects. These edges were obtained by thresholding a local area gradient image derived from the intensity-normalized and spatially filtered image.

The total process consists of a cascade of sub-processes. These are grouped by the traditional three-part pattern recognition terminology of preprocessing, feature extraction and classification. The important point to note is that the throughput rate for the total system is essentially that of the slowest sub-process. As is usual for pipe-line processor design, a conscious attempt is made to divide the processing sequence into sub-processes in such a way as to ensure that the slowest is not much slower than the rest, and, of course, is fast enough. Table 2-1 summarizes these qualities.

A general characteristic of the sub-processes used is that only local neighborhoods are involved in any high-volume computations. This minimizes the amount of storage required. For most operations, only a few scan lines are needed and FIFO registers, possibly implemented with CCD's, could be used.

It should be noted that there are no cognitive processes, also, no feedback is used. Thus the performance of the system depends heavily on correct interpretation and use of a priori knowledge of the scene and sensor. Its success is contingent on the stability of these characteristics, as no learning mechanism is built in. Such considerations are beyond the scope of this report.

As previously mentioned, the data set consisted of 38 images. These were divided into a design set of four images (containing a total of four targets) and a test set of 34 images (containing 43 targets total). The classifier logic was designed on the former (i.e., four images) which included four tactical targets, nine interior boundary traces, 36 residual noise blobs four marker symbols, and 31 "other" shapes. The latter class was the catch-all category. Though the initial objective was only to separate targets from non-targets, the non-targets clustered into well-defined groups based on size and shape. To demonstrate a slightly more interesting classifier at essentially no extra cost, we designed the classifier to distinguish each class from the rest. Actually, by breaking the total problem into a set of smaller and easier sub-problems, the total classifier logic became quite elementary.

The effectiveness of the classification logic was evaluated by applying it against the test set of 34 images. In summary, out of a

Table 2-1

Characteristics of Algorithms Proposed for Cueing

1. <u>Real-Time Potential</u>

   - Suited to "pipe-line" processing.
   - Total processing delay: several frames.

2. <u>Realizable With Straightforward Logic</u>

   - Table-look-up image processing.
   - Few arithmetic operations
   - Simple decision logic.

3. <u>Sub-field Temporary Storage Requirements</u>

   - Target-size sub-field sequential image memory, 2 and 3 scan-line shift register memory, and histogram counters.
   - Possible use of CCD components for sub-field and scan-line memory.
   - Small scratch-pad only random-access memory required.

total of 43 targets, 38 were detected (88% probability of detection). The five targets not detected were all placed in the "other" category. It should be noted that there was only one instance of a non-target being called a target. Moreover, if the superimposed graphics had not been part of the digitized imagery, there would have been no false alarms and only two missed targets.

Details regarding the noise suppression, feature extraction, and classification logic design and evaluation processing are presented in Section 4. Descriptions of the DICIFER system hardware and software configurations are provided in Section 3.

SECTION 3

DICIFER SYSTEM BACKGROUND, PHILOSOPHY, AND FUNCTIONS

Rome Air Development Center (RADC) has developed a digital image processing capability within the Reconnaissance and Mapping Branch (IRR) of the Intelligence and Reconnaissance Division (IR). Full capability now exists for data handling, preprocessing, searching, measurement extraction and evaluation, feature data structure analysis, and recognition logic creation and evaluation for digitized image data from a variety of sources. Two major applications areas are being actively pursued. The first is the development of semi-automatic reconnaissance imagery target screening and recognition procedures. The second is land surface thematic mapping based on multispectral images. The former problem has long been of interest to the intelligence community. The interest in the latter problem stems from new equipment capable of obtaining high-quality registered images, each image representing a small portion of the spectrum, and from the notion that certain classes of earth surface material can be distinguished by their spectral signatures.

3.1. IFES/SCORE DEVELOPMENT

The RADC image processing system [1], known as DICIFER (Digital Interactive Complex for Image Feature Extraction and Recognition), is interactive to allow the researcher/analyst the greatest amount of flexibility in guiding the design of processing logic according to particular characteristics of the data and output requirements. Implementation on a general-purpose computer has allowed for future expansion and modification as desired. The DICIFER acronym represents most of the keywords that describe the system:

- Digital: The data are input, processed and stored in digital form.
- Interactive: The system processes are chosen and continuously guided by human direction.
- Complex: The system includes a comprehensive array of equipment.
- Image: The processing routines and equipment are oriented toward image data.
- Feature: Features (measurements) are defined which, on the basis of a priori knowledge and/or experimental results, are reliable class discriminators.

- o Extraction: Routines exist to extract these features from raw or preprocessed data to build up multi-dimensional vector files to be used as input to classification routines.

- o Recognition: The recognition of targets and the correct assignment of a picture elements, as represented by their respective vectors, are accomplished by the classification logic designed and evaluated on-line.

The hardware consists of a dedicated mini-computer, two disks, magnetic tape, storage display with hardcopy, line printer, and equipment for inputting and quantizing film and printed images and information from multichannel analog tape. Files can be displayed in black-and-white (0-255 grey levels) or pseudocolor, and can be output in the form of color-coded (64 levels) or black-and-white transparency files measuring up to 1024 x 1024 pixels.

The system software capability has been provided by Pattern Analysis and Recognition Corp. (PAR) under a number of contractual efforts with RADC. These efforts were directed at developing the basic operating system and applications-oriented measurement routines. An additional effort provided for the implementation of the OLPARS (On-Line Pattern Analysis and Recognition System [2]) capability on the DICIFER system. OLPARS has been an on-going development at RADC and is resident in many forms both on general-purpose computing systems and in dedicated configurations oriented toward the investigation of a particular problem.

The system description given in the following subsections describes the various processes involved in solving the types of problems mentioned above. These include three broad types of applications routines which do:

1. Image-to-image mappings
2. Feature extraction
3. Classification logic design

Table 3-1 lists specific algorithms on DICIFER. A description of the hardware and software organization is also included.

To summarize, DICIFER is an interactive general-purpose system with the image processing, feature extraction, and logic design capabilities necessary to solve a large variety of problems. Applications of the system to the enhancement of radar imagery, recognition of tactical targets in aerial photography, and classification of multispectral data into land use categories have been investigated. The results achieved thus far indicate that rapid progress can be made toward the solution of a wide class of problems through the use of an interactive system such as DICIFER which contains the necessary variety of tools.

Table 3-1 SYSTEM CAPABILITIES OF DICIFER

Primary Application	Image to Image Mappings	Feature Extraction	Classification Logic Design
General purpose, but oriented towards remote sensed imagery of national interest	Smoothing, weighted smooth(includes Laplacian), odd dot elimination, odd line elimination, point edge enhancement, area edge enhancement, max area edge enhancement, thresholding, range change, closed curve routine, Hadamard and Fourier Transforms and inverse trans., filters, grey-level linearization, add and subtract two images, translation, rol[illegible]picture, decimation, interpolation, scale change, fill-in and adaptive fill-in routine, multispectral image manipulation	Spectral coefficients, average grey level, variance of grey level, number of points with specific range of grey levels(includes area and perimeter measurements), connected component measurements, sub-area search executive routines, spectral map extraction, linear combinations, user add-on capability	Pairwise-Fisher logic, piecewise linear logic drawn on(a) optimal discriminant plane, (b) eigenvector, (c)arbitrary vector, (d)coordinate axis, (e)Fisher direction vector or vector-pair (plane projection); Boolean Compiler to allow logical combination of algebraic expressions to be used in inequalities as decision criteria;Preclustering and Non-linear Mapping algorithm and 1 and 2-space projection for analysis of modality of data classes

3.2. THE GENERAL PATTERN RECOGNITION PROBLEM

Pattern recognition theory grew from an interest in modeling neural behavior and in attempting to imitate by mechanical means the recognition and decision-making functions of man. This interest has not diminished even though a greater appreciation of what can be accomplished with today's techniques now exists.

The general pattern recognition logic design problem is to create the transfer function for a system that will produce the response appropriate to the stimulus. The usual form of response is the generation of a code word identifying the class to which the stimulus is judged to belong. The transfer function is usually a many-to-few mapping, with many more stimulus examples than possible classes. The pattern recognition system achieves a partitioning of the sets of sample points in a high-dimensional pattern space by means of decision boundaries obtained by design. The system may also be asked to decide when to respond, i.e., when it recognizes the presence of a stimulus to which it should respond. Visual stimuli might be present on a page of printed text, an aerial photograph, a set of spectral images, or a photomicrograph of biological cells. Images, or portions thereof, are usually transduced into an equivalent electronic form by digitizing discrete samples spaced at regular intervals for subsequent storage and processing. A search function has the task of isolating the pertinent information which is to be recognized.

The process of generating responses usually involves a sequence of concatenated operations including preprocessing, feature extraction, and classification as well as the raw data transduction and search operations already mentioned. Image-to-image preprocessing transformations are sometimes applied to correct for known systematic geometric and radiometric distortions, to filter out redundant data to enhance certain information for visual presentation and/or to transform the data to make it easier to extract features. These features (or measurement values) are used by the classification logic to accomplish the recognition task if object (or point) classification is desired.

The derivation of a useful set of features is the most critical part of the solution to any pattern recognition problem. The ideal selection criterion is that they possess only the information essential to differentiate objects adequately according to class membership. A set of features for an object (or point) is normally referred to as a feature vector, and it is this vector which is passed on to the classification logic to be recognized as belonging to a class set. The classification logic determines the sub-space in which the feature vector is located, in order to generate the class decision code along with an indication of the confidence in that decision.

The choice of features and of classification logic for a given problem is obtained at present largely by an ad hoc procedure. Feature definition, hence the structure of the sets of feature vectors, is strongly dependent on the application, as well as on prior transformations. Satisfactory solutions to difficult problems seem to require many hours of study and experimentation with large amounts of representative data and a thorough understanding of the underlying physical phenomena which dictate the empirically observable class variations. Furthermore, the tools of the analyst often determine the ease with which a solution is obtained and probably the nature of the solution as well. Figure 3-1 is a flowchart of the typical process for solving pattern recognition problems.

3.3. SYSTEM PHILOSOPHY

DICIFER is intended to be used as a research tool in a highly interactive manner. A researcher would typically try many different algorithm sequences with various parameter settings, evaluating his results on-line until a satisfactory solution is achieved. A general-purpose digital computer provides the necessary flexibility to modify and add routines to the system.

Some algorithms would be too slow for repetitive production work, but their utility is currently measured in terms of other performance criteria, e.g., correct recognition rates. It is anticipated that production versions would employ special computing hardware to enhance speed.

DICIFER is also applicable to problems where the desired response is a modification of the input imagery in which objects to be recognized are enhanced so that an analyst may more easily perform the recognition task. In addition, the system may only be asked to highlight those areas that have simple features, indicating the probable locations of the targets of interest, and then to provide the figure-background separation necessary for object identification.

The methodology used to obtain solutions to pattern classification problems requires that these capabilities reside in one system if convergence to a solution is to be efficiently achieved. It is only through the application of classification logic that the adequacy of a feature set can be ascertained. It is through the process of error analysis that ideas for modifications and additions to the feature set to improve discrimination are generated. When the modifications and additions have been made, new logic must be designed to accommodate this cycle of feature design, logic design, and error analysis.

3.4. FACILITY HARDWARE

The DICIFER hardware system is configured about the following primary components: DEC (Digital Equipment Corporation) PDP-11/20

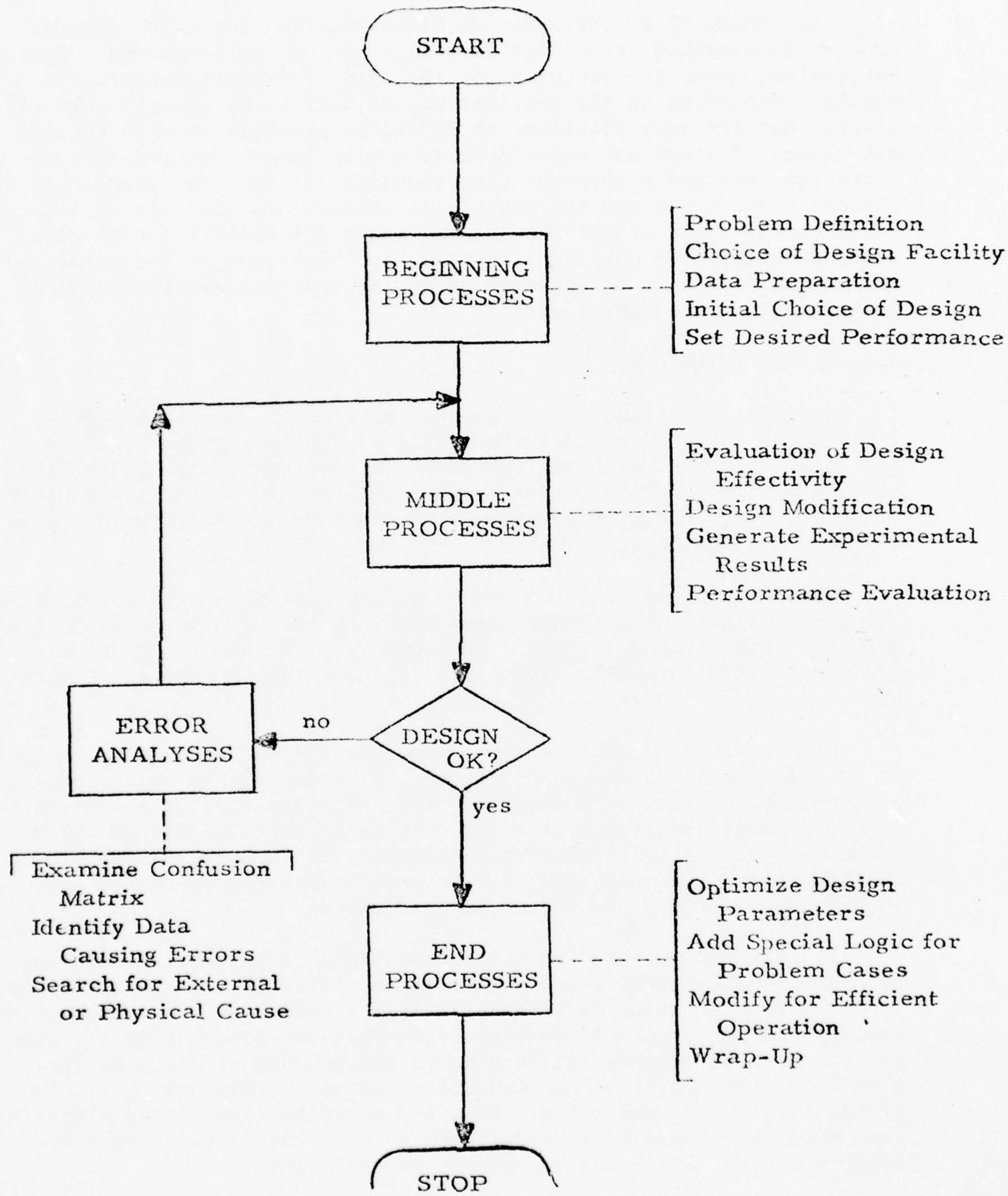


Figure 3-1 Typical Process for Solving Pattern Recognition Problems

Computer System, SDS (Spatial Data Systems) 800 Display System, Tektronix Interactive Terminal, and RADC Color Output Film Printer. A brief description of each component and its function is contained in the following paragraphs.

3.4.1. PDP-11/20 Computer System

Performs all processing, file manipulation, and input/output functions. The system consists of a PDP-11/20 processor with 28,672 16-bit words of core memory (4096 locations are assigned to the Unibus), an Extended Arithmetic Element, RS-11 256K word fixed head disk, RP02 10-million word disk pack unit, TEC-10 seven-track and nine-track industry compatible magnetic tape units, dual DEC tape unit, teletype and card reader. The 10-million word disk pack is used mainly for storage of digitized images. The system routines are stored on the fixed head disk and are called into core by a resident executive program in response to user requests entered via keyboard. User communication with the system is via the Tektronix display terminal, teletype, and/or cursor and monitor.

3.4.2. SDS Input and Display System

Provides for input of image data through a TV-Vidicon Camera (SDS Computer Eye), and image display in B/W and/or color on two CRT's. An image may be viewed directly from the camera prior to digitization or the digital equivalent may be viewed on the display. In the display mode, the system accepts a digital image file from the PDP-11/20 and displays it in color and/or B/W with internal disk refresh. The SDS System 800 configuration is composed of an 804/PDP-11/20 Digital Interface, an 804-2 joystick controlled cursor, a Digital Video Converter Model 804 (digital-to-analog interface for the color display), and a Data Color 703 (color display with 32 color coded levels, 490 rows x 384 columns raster-scanned). A black/white Miratel monitor and refresh storage device (also with 490 rows, 384 columns, with 256 grey-level code) is interfaced to the system and is the primary hardware for viewing and interacting with imagery. Raw video, digitized images, cursor cross hair or any combination of these can be displayed separately or superimposed on this monitor.

3.4.3. The Tektronix Interactive Terminal

The Tektronix Interactive Terminal is a primary instrument of the man-machine interface; it displays software menus and system dialogue as well as graphics which aid in analysis (e.g., histograms and scatterplots). Any information displayed can be retained in hardcopy form. The hardware consists of a Tektronix 4010-1 storage tube display terminal and 4610 heat-processing hardcopy output unit.

3.5. SOFTWARE

Functionally there are three modules of applications software:

1. Preprocessing
2. Measurement Extraction
3. Structure Analysis & Logic Design

These programs are called by the executive routine according to user selection from option menus. They depend heavily on various display routines which allow the analyst to view imagery and to view the results of algorithms which operate on imagery and/or vector data. Figure 3-2 depicts the basic block diagram of the system. Convenient file manipulation and logical executive control are important aspects of the operation of DICIFER. The following briefly describes the design algorithms available.

3.5.1. Preprocessing

In the preprocessing phase, the image is processed to enhance the characteristics of the objects to be recognized or, equivalently, to eliminate information from the image not pertinent to the detection and classification of the objects. The output is another image. Smoothing, noise elimination, edge enhancement, and "line manipulation" algorithms (which operate on edge-detected images) comprise the "local" mapping in the preprocessing module. These are local mappings in the sense that the grey level at a point in the preprocessed image is a function only of grey levels at points in the neighborhood of the corresponding point in the original image. This distinguishes them from the Fourier and Hadamard transformations and filters, which are global in nature and which are also contained in this module. Because the output of any of the preprocessing algorithms is in image format, many preprocessing options may be concatenated by using the output image of one algorithm as the input image to the next algorithm, etc., until the desired result (enhancement) is achieved. A detailed discussion of the preprocessing routines and their application to reconnaissance imagerv may be found in References [1,7, and 8].

3.5.2. Measurement Extraction

The search routines are applicable when the target object is small in relation to the size of the image. These translate a rectangle to various positions within the image, applying specific search criteria to picture elements in the rectangle. A criterion may be based on a masking operation, texture measurements and/or topological measurements. The positions of the rectangle which satisfy the criterion are noted and the enclosed "areas of interest" define area files (sub-region boundaries) which can be called for further processing.

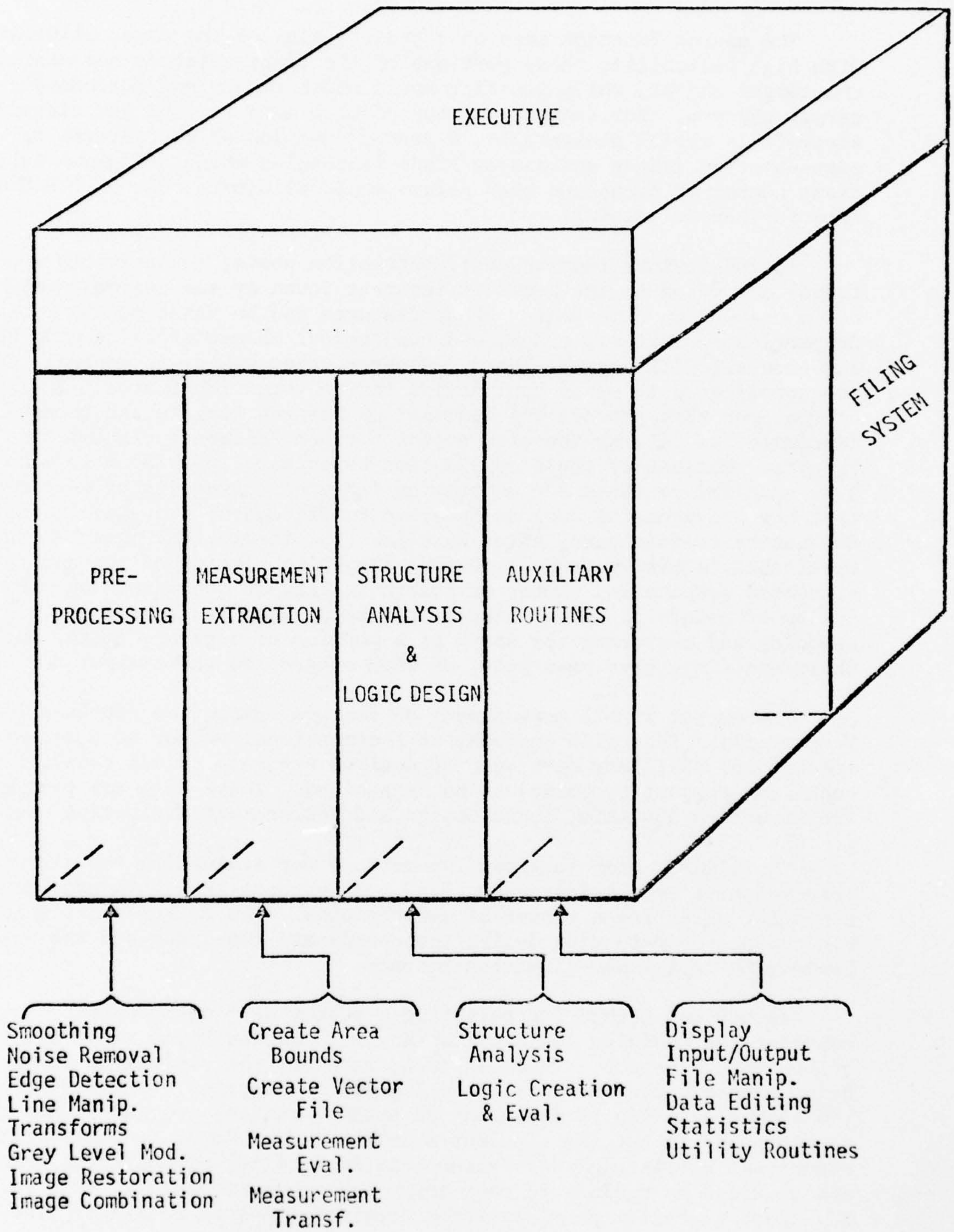


Figure 3-2 Building Blocks of Software System

The search function acts as a prescreening of the data, eliminating with high reliability those portions of the image which do not contain the target object, while locating areas which may or may not contain target objects. For example, if the problem were to find and classify aircraft in aerial photography, a search function which operates on edge-detected images and simply finds rectangles which contain a sufficient number of connected edge points would eliminate most of the image from further processing.

In the feature (measurement) extraction phase, measurements (features) are taken on the areas of interest found by the searching algorithm or in user-drawn sub-areas. These features can be based on the same properties of texture, shape, and topological characteristics upon which a search algorithm may be based. However, they would, in general, be too complicated to merit application to the total image area. The set of features must provide the distinction between targets and target facsimiles as well as the distinction between different classes of targets. Because of their application dependence, provision is made for user-supplied routines for extracting features. Examples of features that may prove useful include Hadamard coefficients; grey-level spatial dependency coefficients, after Haralick [3]; topological measurements applicable to binary (edge-detected) images consisting of the number of connected components, number of points in largest connected component, number of points in second largest connected component, etc. Methods of encoding and measuring the shape of a portion of a binary image, such as chain encoding, have been left, as with others, to the analyst to define.

The output of all measurement or feature extraction routines is a Vector File. This file contains an L-dimensional vector of L measurement values, and ancillary data such as desired response (class code), for each item (object or point) to be recognized. These data are used by the Structure Analysis, Logic Design and Measurement Evaluation routines.

The DICIFER user is given two methods for evaluating the discriminatory value of each measurement [2,4]. In essence they both provide a means for selecting a subset of measurements. The measurements which are chosen for retention define the coordinate sub-space and the desired projection to a lower-dimensional space.

An optimal method for selecting a subset of M measurements must consider the decision logic criterion, such as the Bayes risk or the probability of error. This, in turn, requires the estimation of the joint probability functions for all possible n-tuples. The computational difficulties in obtaining an optimal ranking precludes this approach in all but the simplest problems. Therefore, the sub-optimal algorithms of Discriminant Measure and Probability of Confusion Measure are provided as options to rank order the L measurements x_1, x_2, ..., x_L. Each algorithm provides three distinct types of rankings. The

first uses a significance measure for a particular component, say x_p, for discriminating class i from class j. This significance will be designated as $M_{ij}(x_p)$. The second type of ranking uses a significance measure of x_p for discriminating class i from all other classes, and is designated $M_i^p(x_p)$. The last type of ranking uses a measure of the overall significance of x_p for discriminating all classes and is designated $M(x_p)$. A discussion of how the evaluation is used is in Section 4.7 of reference [1].

3.5.3. Structure Analysis and Logic Design and Evaluation

In the structure analysis phase of the logic design, the analyst attempts to determine how the vectors from the different classes are distributed in the L-dimensional feature space.

Parametric techniques of pattern recognition theory assume that the data from each class are unimodal and distributed according to a particular probability distribution, such as multi-variate Gaussian. In real-world problems, however, such assumptions are risky, and may lead to poor results.

Structure analysis assists the analyst in learning the modality of each class. The vectors which belong to a mode (single class cluster) can be relabelled and treated as sub-classes which are later recombined by the decision logic.

Projection of the data onto the 2-space spanned by the two eigenvectors of the lumped covariance matrix which have the two largest eigenvalues is one classical method for determining structure of multi-dimensional data. A mode can be separated from the rest of the vectors by drawing a piecewise-linear convex boundary around it on the display, and specifying a new label for the vectors represented by the points within that boundary. Similar routines exist for projecting histograms on a single vector. The one- or two-space vectors may also be arbitrary, original feature coordinates, or Fisher vectors instead of being eigenvectors. For example, an "Optimal Discriminant Plane" projection first projects points onto a Fisher direction (for a chosen pair of classes) and then onto the Fisher direction orthogonal to the original.

The outlines of the rectangles from which these vectors were extracted may also be displayed against the background of the image from which they came, thus allowing the analyst to see the specific areas of the image which comprise the particular mode. This may allow him to perceive the physical reason for the mode to exist. The desire to be able to relate regions of the L-dimensional feature space back to the original source information was the primary motivation for adding OLPARS display capabilities to the DICIFER system.

A powerful non-classical technique for structure analysis included in the system is the Non-Linear Mapping algorithm [5]. The algorithm calculates a mapping of points in the L-dimensional feature space to a 2-dimensional space while attempting to preserve interpoint distances. It includes a preclustering routine to reduce the number of points to manageable levels. A detailed description of Non-Linear Mapping and eigenvector projection is in Section 4.21 of [1].

The decision logic for implementing the desired classification must be based on the set of vectors available at the time of design. If one is fortunate enough to have continual access to new data, it is desirable to be able to update the decision logic if necessary. The redesign may affect the entire logic of some types of classifiers, which may require substantial effort to accomplish. If performance does not extrapolate well to a particular pair of classes, it is convenient to be able to upgrade its response to those classes specifically without affecting other classes.

A hierarchial structure is used in DICIFER to allow easier upgrading of the classifier when needed. It also allows the total problem to be divided up into smaller problems which are, perhaps, easier to solve. This structure also follows naturally from the fact that many classification schemes developed in the past have specialized in two-class dichotomies. The process of obtaining a response to a given vector input can and often does involve a sequence of partial decisions. Thus, the structure of the decision logic is tree-like, with branches and nodes. The Logic Creation and Evaluation module includes, but is not limited to, the following types of logic which the analyst may call upon for use at any node of the decision tree: Fisher Pairwise, 1-and 2-space projections, and boolean. The system keeps a record of the decision logic as it is being designed. The record determines the classification procedures for the recognition of the new data.

Fisher discriminant logic first calculates the Fisher direction vector, $\vec{d}_{ij}$, and a threshold, θ_{ij}, for each pair of classes A_i and A_j in the data base [6].

Classification of an unknown vector, $\vec{V}$, is done on a pairwise basis. For each pair of classes A_i, A_j, the following decision procedure is adhered to: if $\vec{V} \cdot \vec{d}_{ij} \geq \theta_{ij}$, a vote counter for class A_i is incremented. If $\vec{V} \cdot \vec{d}_{ij} < \theta_{ij}$, a vote counter for A_j is incremented.

When all pairwise tests have been completed, the class with the maximum number of votes is taken as the decision. If two or more classes are tied for the maximum number of votes, the vector $\vec{V}$ is assigned to a reject class R as its decision class. Several other reject criteria are also possible.

There are several kinds of plane projection logics, but all require that two vectors in the L-dimensional space be chosen. They may be two eigenvectors of the lumped covariance matrix, the vectors from the Optimal Discriminant Plane [6], a pair of Fisher directions, two coordinate axes, or arbitrary vectors designated by the user. The data are projected on each vector and this pair of values determines coordinates of points in a 2-dimensional plane displayed on the CRT, the identity of each vector being indicated by the class symbol of that vector. The analyst may now draw (by designating end points of line segments with the aid of a cursor) several piecewise linear boundaries to separate classes or groups of classes from one another on the display. The analyst may also designate a region as a reject region. The logic will classify an unknown vector, $\vec{V}$, by projecting it onto the plane and determining into which region it falls. The decision at this point may be only a partial decision, because such a region may contain more than one class. Logic can also be designed using projections on single vectors.

The boolean logic option allows the analyst to write decision criteria in the form of boolean/algebraic statements on the coordinates of the feature vector. Any meaningful statement involving the algebraic operations of addition, subtraction, multiplication or division; integer constants; the equality and inequality symbols; or the logical connectives conjunction ($\wedge$) and disjunction ($\vee$), may be written. A dichotomy is achieved based on the truth or falsity of the predicate.

A feature of DICIFER (not found in previous OLPARS installations) is the ability to evaluate the logic at each node before proceeding with other node logic designs. Formerly, the entire tree had to be completed before evaluation could take place. It was considered important to allow the analyst to detect and correct deficiencies as they were made.

A diagram of the framework of DICIFER is given in Figure 3-3 and a complete description of the individual options may be found in [1].

Figure 3-3. DICIFER

PATTERN ANALYSIS & RECOGNITION CORP.
ON THE MALL
ROME N Y 13440

FRM 0-MASTER FRAME

1-INPUT
2-DATA EDITING
3-STATISTICS
4-PREPROCESSING
5-SEARCH
6-MEASUREMENT EXT
7-MEASUREMENT EVAL
8-MEAS TRANSFORMATIONS
9-STRUCTURE ANALYSIS
10-CREATE LOGIC
11-FILE MANIPULATIONS
12-UTILITY FUNCTIONS
13-DISPLAY OPTIONS

PREPROCESSING

FRM 1-INPUT

1-SDS INPUT-U
2-SDS INPUT-D
3-DISK TO TAPE
4-TAPE TO DISK
5-TAPE DIRECTORY
6-CHANGED TAPE
7-INITIALIZE TAPE
8-LIPS TAPE INPUT
9-COLOR PRINTER
10-SINGLE IMAGE
11-DISPLAY OPTIONS
12-MASTER FRAME

FRM 3-STATISTICS

1-DENSITY
2-MIN DIF
3-MAX DIF
4-AVE DIF
5-DENSITY-AREA
6-MIN DIFF-AREA
7-MAX DIFF-AREA
8-AVE DIFF-AREA
9-PERCENT
10-SPECTRAL MAP
11-COMP SPEC MAP
12-MULTI COMP SPEC MAP
13-BINARY COMP SPEC MAP
14-MISC STATISTICS
15-SINGLE IMAGE
16-DISPLAY OPTIONS
17-MASTER FRAME

FRM 2-DATA EDITING

1-RECT AREA SELECT
2-RECT AREA SEL-SPEC
3-CURSOR RECT AREA SEL
4-CUR RECT AREA SEL-SPEC
5-DRAW AREA BOUNDS
6-BLOW UP DRAW AREA
7-AREA COORDS
8-BOX AREA
9-PICK REG POINT
10-REG IMAGES
11-IMAGE REGIST
12-DISPLAY X-Y
13-BLOW UP IMAGE
14-IMAGE TO SPEC
15-SPEC TO IMAGE
16-INIT DSGN VECT FILE
17-INIT TEST VECT FILE
18-SINGLE IMAGE
19-DISPLAY OPTIONS
20-MASTER FRAME

FRM 4-PREPROCESSING

1-AREA EDIT
2-THRESHOLD
3-NORMALIZE
4-ELEMENT CHANGE
5-RANGE CHANGE
6-SHIFT PIC
7-DECIMATION
8-SMOOTH
9-NOISE ELIMINATION
10-LINE MANIPS
11-COMBINE IMAGES
12-TRANSFORMS
13-SINGLE IMAGE
14-DISPLAY OPTIONS
15-MASTER FRAME

FRM 15-SMOOTH

1-BOX SMOOTH
2-NXM SMOOTH
3-WEIGHTED SMOOTH
4-WGHTD SMOOTH-ABSOLUTE VAL
5-SINGLE IMAGE
6-DISPLAY OPTIONS
7-RETURN

FRM 18-COMBINE IMAGES

1-ADD IMAGES
2-DIFFERENCE
3-ABSOLUTE DIFFERENCE
4-DIFF FACTOR
5-RATIO IMAGES
6-SPECTRAL SIGNATURE
7-SINGLE IMAGE
8-DISPLAY OPTIONS
9-RETURN

FRM 16-NOISE ELIMINATION

1-ODD DOT ELIMINATION
2-ODD LINE ELIMINATION
3-SINGLE IMAGE
4-DISPLAY OPTIONS
5-RETURN

FRM 19-TRANSFORMS

1-HADAMARD TRANSF
2-HADAMARD FILTER
3-FOURIER TRANSF.
4-INV. FOURIER TRANSF
5-SINGLE IMAGE
6-DISPLAY OPTIONS
7-RETURN

FRM 17-LINE MANIPULATIONS

1-POINT EDGE DETECTION
2-AREA EDGE DETECTION
3-AREA EDGE DETECTION-MAX
4-SIMPLE FILL IN
5-ADAPTIVE FILL IN
6-CLOSED CURVE
7-EDGE PRINT
8-TEK EDGE DISPLAY
9-SINGLE IMAGE
10-DISPLAY OPTIONS
11-RETURN

FRM 20-MISC STATISTICS

1-LOCAL STATISTICS
2-DETAILED LOCAL STATS
3-DISTANCE
4-DIST-D
5-DIST-U
6-POINT CNT
7-IMAGE MASK
8-SINGLE IMAGE
9-DISPLAY OPTIONS
10-RETURN

MEASUREMENT EXTRACTION

FRM 5-SEARCH

1-RECTANGULAR SEARCH
2-SINGLE IMAGE
3-DISPLAY OPTIONS
4-MASTER FRAME

FRM 7-MEASUREMENT EVAL

1-SELECT DATA SET
2-PAIRWISE HISTOGRAMS
3-MULTI HISTOGRAMS
4-PROB OF CONF
5-DISCRIMINANT
6-SINGLE IMAGE
7-DISPLAY OPTIONS
8-MASTER FRAME

FRM 6-MEASUREMENT EXT

1-SPECTRAL MEAS-DSGN SET
2-SPECTRAL MEAS-TEST SET
3-CREATE MEAS. SET
4-MEAS. TO VECTOR
5-ADD MEAS. TO VECTORS
6-OUTPUT VECTORS
7-SINGLE IMAGE
8-DISPLAY OPTIONS
9-MASTER FRAME

FRM 8-MEAS TRANSFORMATIONS

1-SELECT DATA SET
2-MEAS TRANSFORM-DESIGN SET
3-MEAS TRANSFORM-TEST SET
4-SINGLE IMAGE
5-DISPLAY OPTIONS
6-MASTER FRAME

STRUCTURE ANALYSIS & LOGIC DESIGN

FRM 9-STRUCTURE ANALYSIS

1-SELECT DATA SET
2-ONE SPACE-HISTOGRAM
3-TWO SPACE-CLUSTER PLOT
4-NONLINEAR MAP
5-UNSTRUCTURE
6-SINGLE IMAGE
7-DISPLAY OPTIONS
8-MASTER FRAME

FRM 10-CREATE LOGIC

1-SELECT DATA SET
2-SELECT NODE
3-DISPLAY CUR TREE
4-MEAN-COV MATRIX
5-CREATE FISHER LOG
6-CREATE 1-SPACE LOG
7-CREATE 2-SPACE LOG
8-CREATE BOOLEAN LOG
9-LOGIC EVAL
10-EVAL TEST SET
11-DELETE LOGIC
12-LIST MEAN AND COV
13-LIST LOGIC
14-CONFUSION MATRIX
15-FLICKER DISP
16-DECISION IMAGE
17-VECTOR COUNT
18-SINGLE IMAGE
19-DISPLAY OPTIONS
20-MASTER FRAME

AUXILIARY ROUTINES

FRM 11-FILE MANIPULATIONS

1-DELETE FILES
2-RENAME FILES
3-DIRECTORY LISTING
4-USER PROG. DIR.
5-INITIALIZE
6-ADD USER FUNCTION
7-CALL USER FUNCTION
8-SINGLE IMAGE
9-DISPLAY OPTIONS
10-MASTER FRAME

FRM 13-DISPLAY OPTIONS

1-SINGLE IMAGE-DIST
2- -UNDIST
3- -INTERP
4- -BLOW UP
5-SIDE BY SIDE-DIST
6- -UNDIST
7- -INTERP
8- -BLOW UP
9-TOP AND BOTTOM-DIST
10- -UNDIST
11- -INTERP
12- -BLOW UP
13-DISPLAY AREA
14-AREA LINES
15-AREA LINES-IMAGE
16-CLEAR GRAPHICS
17-FLICKER DISPLAY
18-BINARY FLICKER
19-DISPLAY UTILITY FUNCTS
20-RETURN

FRM 12-UTILITY FUNCTIONS

1-DOS
2-SINGLE LINE HISTO
3-HARDCOPY
4-GENERATE TEST FILE
5-TEST SDS-800 DISK
6-DIRECTORY AID
7-OCTAL WORD RP02 DUMP
8-OCTAL BYTE
9-DECIMAL WORD-UNSIGNED
10-DECIMAL WORD-SIGNED
11-DECIMAL BYTE
12-USAG11 DUMP
13-CHAINING CHECK
14-CALL ROLLIN
15-SINGLE IMAGE
16-DISPLAY OPTIONS
17-MASTER FRAME

FRM 14-DISPLAY UTILITY FUNCTIONS

1-PICTURE ON
2-PICTURE OFF
3-GRAPHICS ON
4-GRAPHICS OFF
5-NEG PIC
6-ROLL
7-PICTURE PRINT
8-GRID-U
9-GRID-D
10-RETURN

SECTION 4

TECHNICAL DISCUSSIONS

The target cueing technique reported herein has been set up as a sequence of processing steps. These include the following:

- o Noise Suppression
- o Contrast Enhancement
- o Target Boundary Detection
- o Boundary Chain Encoding
- o Feature Extraction
- o Classification Logic Design and Evaluation

The flow diagram of Figure 2-1 provided an overview of these subprocesses. An explicit account of the investigations concerning each is presented in this section. Working descriptions of appropriate algorithms will be provided throughout the text to improve understanding.

4.1. DATA SELECTION *

Due to the volume of imagery received from NVL, it was desirable to establish procedures for referring to the individual images. This information is presented here as an aid in comparing results from parallel efforts.

Thirteen magnetic tapes, labelled A, B, C, ..., M were received from NVL. Each tape contained ten FLIR images. These images have been referred to, during this project, as frames AAAA01-10, BBBB01-10, etc. Each frame was made up of 800 scan lines with each scan line containing 1024 picture elements. Tapes I, J, and M were not usable due to tape errors.

Although the data marks (fiducials, time, altitude, etc.) are not part of the scanned imagery, they are an integral part of the digital images and, as such, contaminate the image statistics. For this reason, we elected to work with that portion of each image which was between the upper and lower data blocks. Specifically, the central 512 rows (scan lines) of each frame were used. Moreover, there appeared to be little usable data in about the first 174 columns of each frame. This space contained a blank area and a region of apparently serious "ringing" transients which appear to be a function of the particular sensor and probably could be corrected through adjustment of hardware electronics. For this reason the first 174 columns of each image were not used. That is, from the ten images AAAA01-10, for example, we selected (512 x 850 pixels) images which are referred to as AAAA01-10, respectively. Sets B, C, etc. were edited in the same manner.

From the available images, 38 frames were selected for analysis during this effort. These frames were selected to be of similar scale but representative of the variety of image quality and target combinations present. The frames selected and the targets contained within the 512 x 850 array are listed in Table 4-1.

---

* Additional comments regarding the data quality assessment and data selection phase of this effort are included in Appendix C.

FRAME NUMBER	TARGETS
B00004	Tank/S
B00005	Tank/S
B00006	Tank/S
B00007	Tank/S
B00008	2-1/2 Ton Truck/E
C00001	APC/S; Tank/ E
C00002	Tank/ E
C00004	APC/S; Tank/E ; 2-1/2 Ton Truck/E
D00007	APC/ E
D00008	APC/ E
E00001	2-1/2 Ton Truck/E
E00002	APC/E; Tank/(3/4 view)
E00003	Tank/(3/4 view)
E00004	No Target
E00005	Tank/(3/4 view)
E00006	APC/E; Tank/(3/4 view)
E00007	2-1/2 Ton Truck/E
E00008	APC/E; Tank/(3/4 view)
H00005	APC/E; Tank/(3/4 view)
H00006	APC/E; Tank/(3/4 view)
H00007	2-1/2 Ton Truck/(3/4 view)
H00008	Tank/(3/4 view)
H00009	APC/E; Tank/(3/4 view)
H00010	APC/E; Tank/(3/4 view)
K00004*	2-1/2 Ton Truck
K00005	Tank/(3/4 view)
K00006	APC/E
K00007	APC/E
K00008	APC/E
K00009	Tank(3/4 view)
K00010	2-1/2 Ton Truck
L00004*	2-1/2 Ton Truck
L00005*	Tank(3/4 view)
L00006*	APC/E
L00007	APC/E
L00008	APC/E
L00009	Tank(3/4 view)
L00010	APC/E

Table 4-1 Images and Targets Making Up Data Set
The four asterisked frames make up the design set, while the remaining frames comprise the test set.

4.1.1. Image Quality

The FLIR images supplied generally used a relatively small portion ($< 50\%$) of the full grey-scale range (256 levels) available. This suggested the use of contrast expansion techniques in order to make better use of the display dynamic range and to achieve visual enhancement.

The range expansion techniques mentioned above also tend to amplify noise within the image. The effect of the noise amplification is to mask partially certain of the targets in the imagery and to make most of the contextual information indistinguishable. It was also determined that certain image noise disrupted to an extent the effective use of the target boundary detection algorithms.

There were at least seven types of noise, loosely described as follows:

1. salt and pepper (single pixel, high contrast),
2. ripple (a high-frequency modulation along most scan lines),
3. ringing (a strong initial transient response on each line),
4. streaking (the absence of scene information on certain lines),
5. herringbone (a high contrast interference pattern),
6. undulation (a low-frequency base line shift), and
7. random (thermal and/or detector noise, etc.).

4.2. IMAGE NOISE ATTENUATION

It was found that noise types (1) and (2) were both bothersome from a visual observation and a gradient detection standpoint and amenable to simple filtering. Consequently, the image data was processed using algorithms designed to suppress these types of noise. The frames (for K00010) shown in Figure 4-2 through 4-5 are used to illustrate results.* Figures 4-6 through 4-9 show a portion of each image displayed without the picture reduction (resampling at integer spacing) used in Figures 4-2 through 4-5.

The processing carried out here results in smoothing of the original data and picture reduction. These techniques can be applied here because the original data was highly oversampled (~2.5 x the Nyquist rate). If this had not been the case, then the tactical targets would have subtended fewer picture elements with the effect that the noise attenuation techniques would degrade (or remove) them also.

---

* The frames shown in this report were photographed as displayed on the DICIFER TV monitor.

4.2.1. Odd-Dot Removal

The salt and pepper noise was removed by the application of a simple non-linear spatial filtering routine applied to a 3 x 3 neighborhood about each picture element, called "Odd-Dot Removal" on DICIFER. Specifically, the difference between the grey value of a given picture element pixel and the average grey value of its eight adjacent neighbors was calculated. For each pixel for which a certain threshold was exceeded, the grey value of that element was changed to the average of its neighbors. A threshold value of ten was utilized. The frames shown in Figures 4-3 and 4-7 represent the result of this processing.

The effects of this and subsequent processing may also be seen by examining the scan line profiles presented in Figure 4-10. These profiles show intensity as a function of position across the scan line indicated between the arrows in frame K00010.

4.2.2. Removal of High-Frequency Ripple Effect

Image analysis indicated a type of noise which was present in most of the imagery. This was, a characteristic fluctuation of grey levels across each scan line, which seems to be the result of a FLIR system functioning problem rather than a result of the actual ground temperatures. The frequency of this "ripple" effect is fairly uniform along each scan line throughout the image; however, phase correlation tends to last for only a few lines.

The fluctuations in amplitude have a noticeable pattern whereby peak-to-peak distances are generally 5 to 7 pixels. In order to compensate for this effect, a simple linear filter was devised which averaged alternating peaks and valleys along pairs of scan lines, using DICIFER's "Weighted Smooth" function. That is, an image which had been processed by the Odd-Dot Removal is next processed using the weighted array shown below:

$$\text{weighted array} = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \end{bmatrix}$$

The result of this processing is to attenuate the amplitude of the fluctuations and to provide an image in which target-to-background contrast ratios appear improved. Moreover, the resulting images were considerably enhanced for viewing purposes, as they presented better target definition and improved contextual background displays.

The frames shown in Figures 4-4 and 4-8 represent the results of this type of spatial filtering.

4.3. CONTRAST ENHANCEMENT

As has been previously indicated, the digitized FLIR imagery utilized only a portion (< 50%) of the available system dynamic range. The effect is that the images, when viewed on the TV monitor, have a flat appearance. Adjusting the gain and bias from frame to frame is useful for improving image appearance. So as not to burden the already busy operator of future FLIR exploitation systems further, such adjustments should be automatically implemented. To accomplish such adjustments, we have used functions available on DICIFER. That is to say, dynamic range expansion techniques have been implemented which modify the input images with changes in gain and bias, slight clipping at the darker end of the image histogram, and some saturation among the brightest image points.

The transfer function utilized to perform this type of range change is schematically illustrated in the diagram of Figure 4-1.

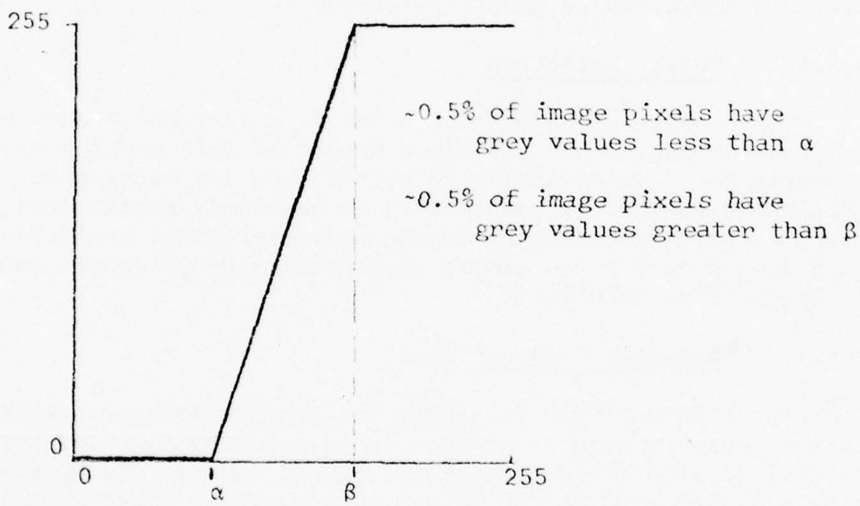


Figure 4-1. Transfer Function Used for Range Change

The effects of the range change processing are to enhance the contrast so as to display the spatial distribution of target intensity more clearly and consistently. Target-to-background contrast ratios are generally improved in the output image.

The frames shown in Figures 4-5 and 4-9 represent the result of applying the range change algorithm to the spatially filtered version of K00010. The image histograms (frequency of occurrence vs. grey level) shown in Figures 4-4 and 4-5 provide for a quantitative comparison of before and after processing examples, as do the respective profiles of Figure 4-10.

4.4. COMMENTS ON PREPROCESSING

The algorithms applied to the FLIR imagery to achieve noise attenuation and contrast enhancement have been described and illustrated. A second example of this processing is illustrated for frame C00004 in Figures 4-13 through 4-20.

As a result of the preprocessing implemented here, most of the salt-andpepper and the "ripple" type noise was suppressed, and target-to-background contrast was normalized. A more complex filtering technique might have been even more effective. However, simplicity of implementation considerations were heavily weighed, and further improvements would not have significantly influenced subsequent processing. For this reason, the preprocessing steps described here were applied to each of the 38 images in the data set prior to the application of boundary detection and encoding processes. These processes will be described in the following subsections.

4.5. INTRODUCTION TO TARGET DETECTION

4.5.1. Object Detection

The problem of object detection is a very broad one, with no single solution or approach. The first aspect of this problem was to select criteria for distinguishing an object from its background. The general foreground-background question is an extremely subtle one, since there are many potential distinguishing characteristics available. Fortunately, most images have a few simple characteristics which are sufficient for distinguishing an object.

4.5.2. Approaches Investigated

The FLIR data was carefully analyzed to determine which characteristics could be used to obtain a good object-background separation. Several different approaches were investigated. One approach used local area statistics to decide if subregions of the image contained target-sized objects. This approach is discussed in Section 5. Another approach investigated was the use of the brightness level (or grey value) of objects as the discriminating characteristic. The targets in the FLIR data are, on the average, brighter than their backgrounds; however, the use of level slicing or brightness thresholding was not an effective method due to variations in overall brightness levels throughout and between images. The approach finally chosen for object detection used the brightness difference between object and background. This brightness difference is detected as a grey-level gradient at the edge or boundary of an object. This choice was also influenced by the availability of gradient detection routines on DICIFER.

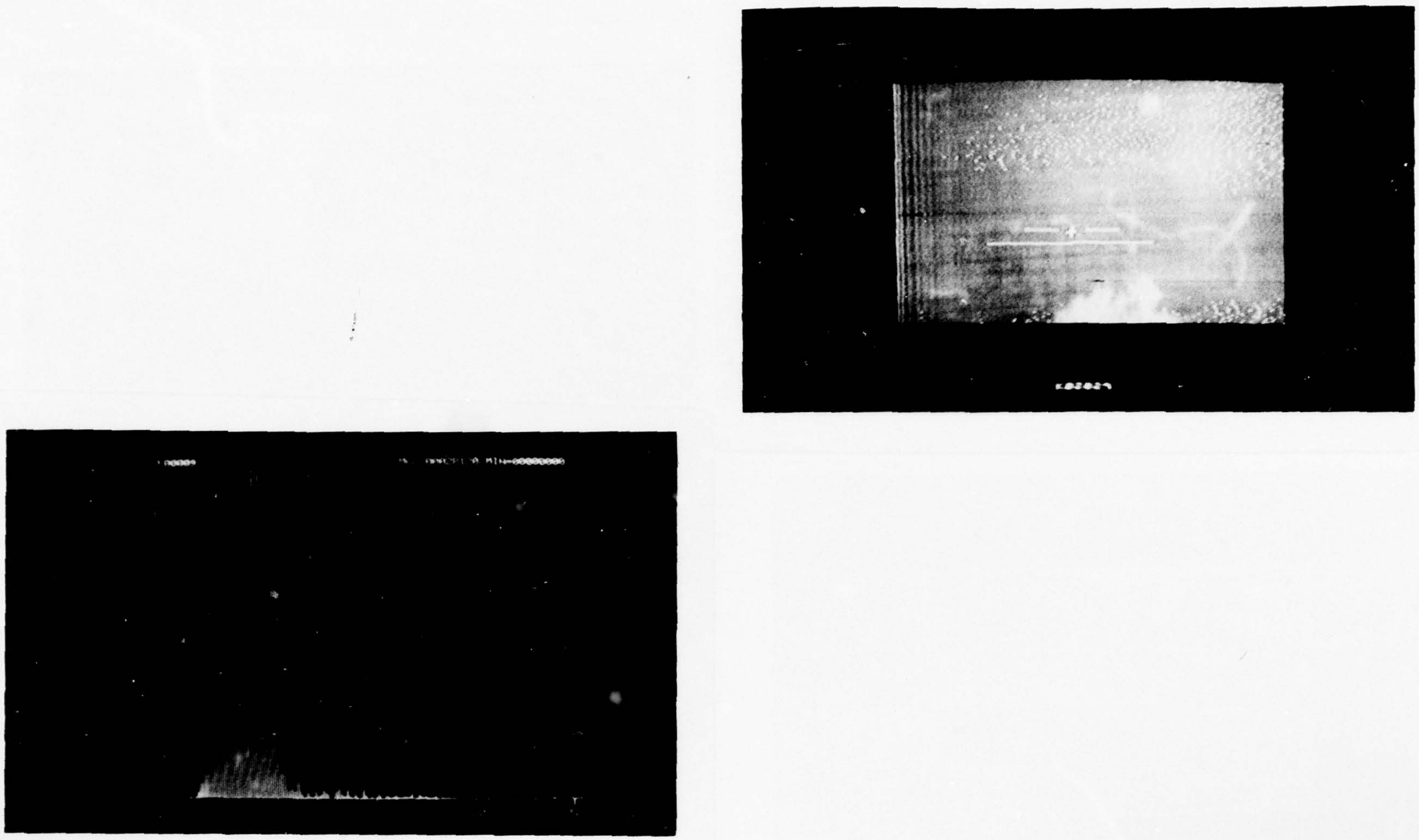
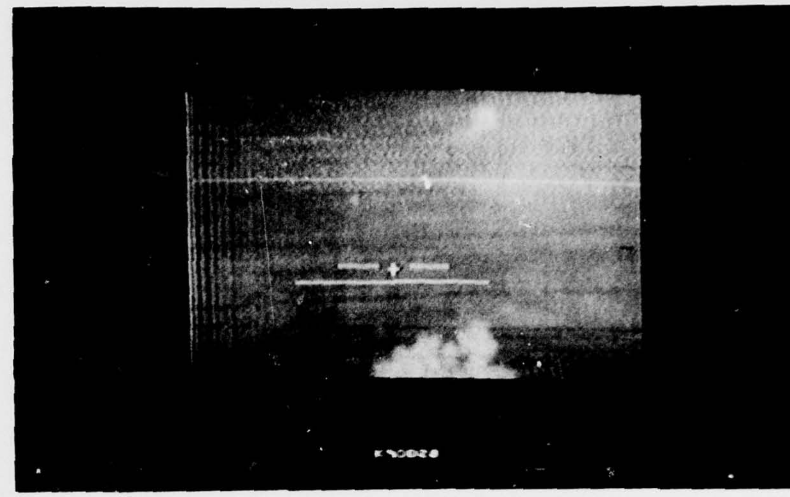


Figure 4-2 Frame K00010 as Displayed on the DICIFER TV Monitor and the Histogram (plot of frequency of occurrence vs. grey level) for this Image. The Target in View is a 2-1/2 Ton Truck. Displayed Image has been Reduced via Pixel Deletion.



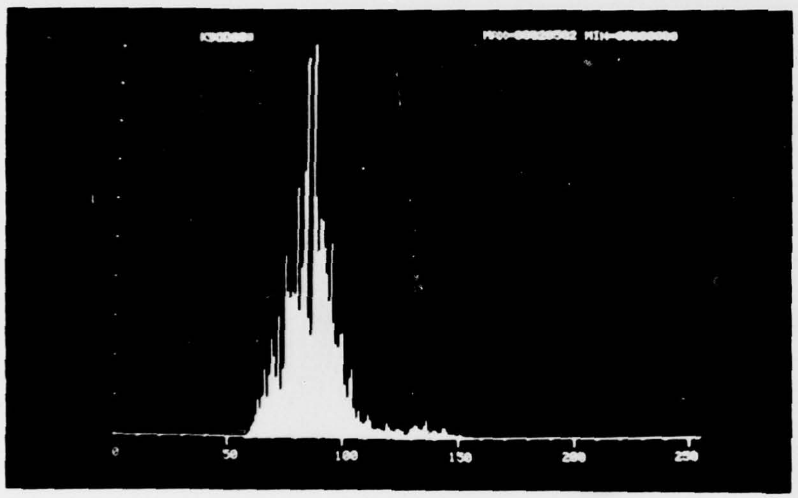


Figure 4-3 Frame K100D0 and Corresponding Histogram.
Results of Processing Frame K00010 for Removal of Salt-and-Pepper
Noise. Displayed Image has been Reduced via Pixel Deletion.

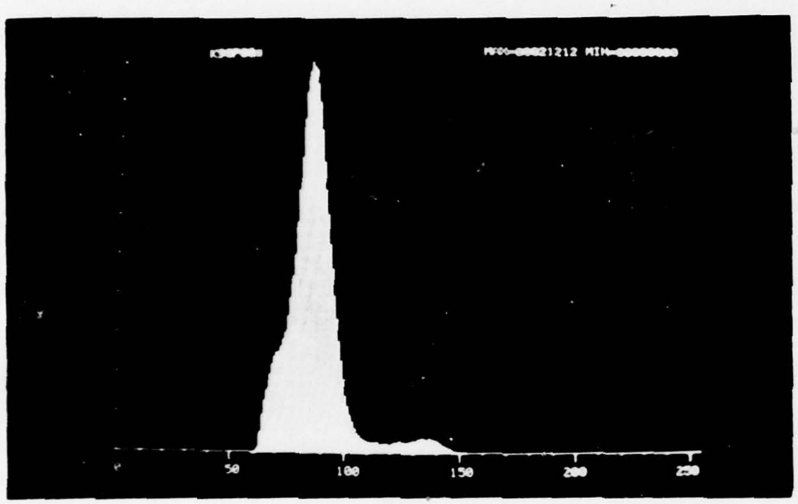
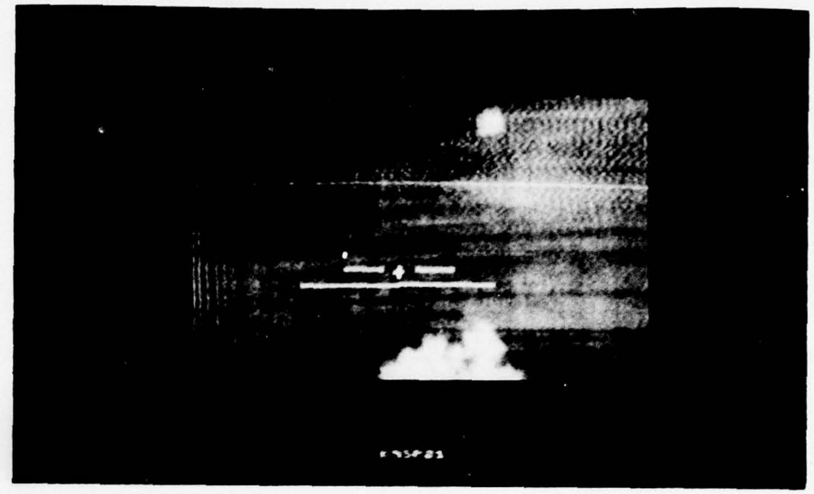


Figure 4-4 Frame K10SP0 and Corresponding Histogram.
Results of Applying Spatial Filtering to Frame K100D0.
Effect is to Suppress High-Frequency "Ripple" Noise Pattern Along
The Scan Lines. Displayed Image has been Reduced via *Pixel Deletion*.



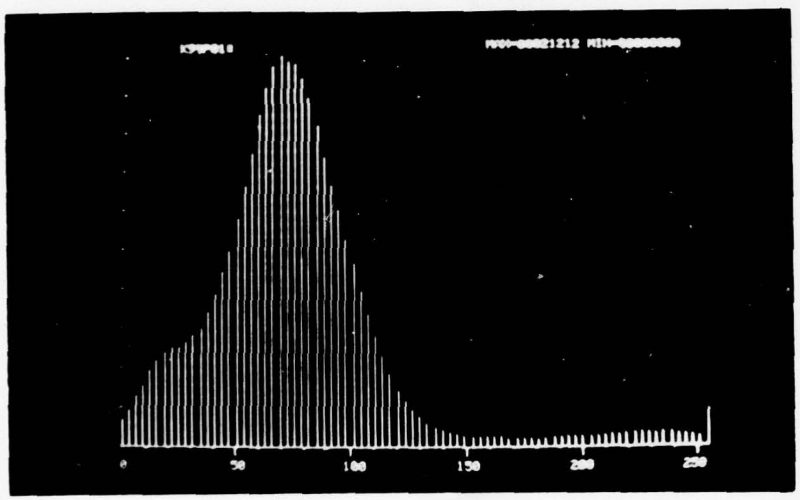


Figure 4-5 Frame K10SP1 and Corresponding Histogram.
Results of Applying Dynamic Range Change to Frame K10SP0 in Order to Achieve Contrast Enhancement. Displayed Image has been Reduced via Pixel Deletion. The apparent missing grey levels in the histogram are a function of rounding-off to integer values within the range change operation.

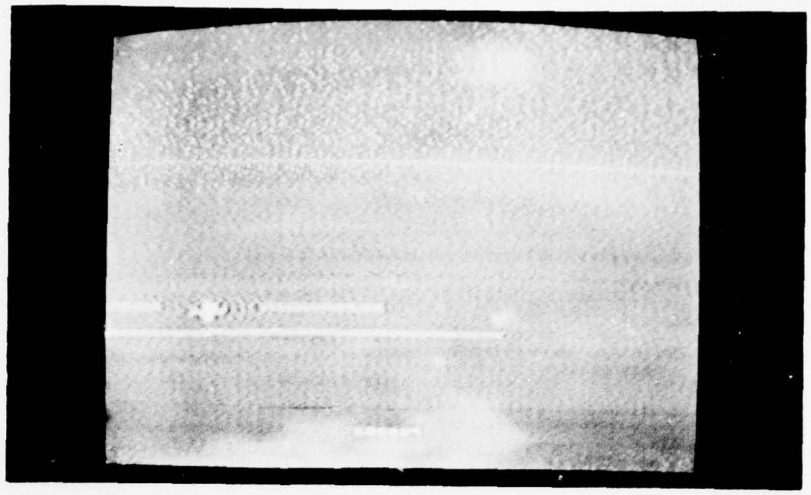


Figure 4-6 Central Portion of Frame K00010 (Figure 4-2) as Viewed when Displaying Every Picture Element.

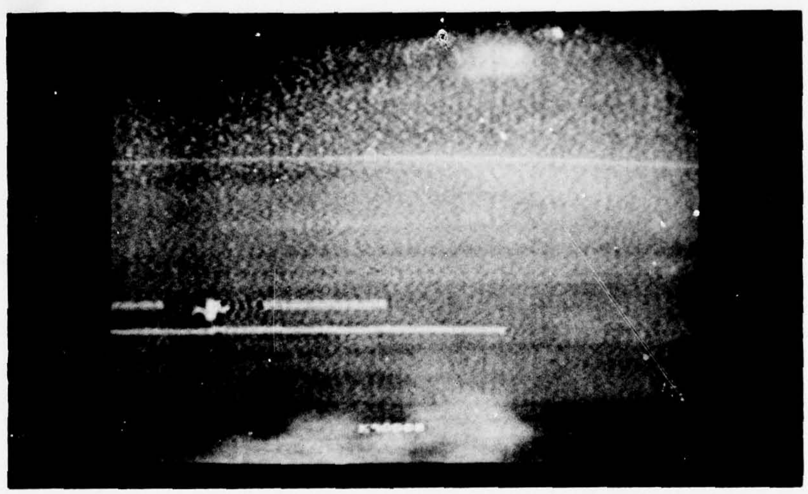


Figure 4-7 Central Portion of Frame K100D0 (Figure 4-3) as Viewed when Displaying Every Picture Element.

Figure 4-8 Central Portion of Frame K10SP0 (Figure 4-4) as Viewed when Displaying every Picture Element.

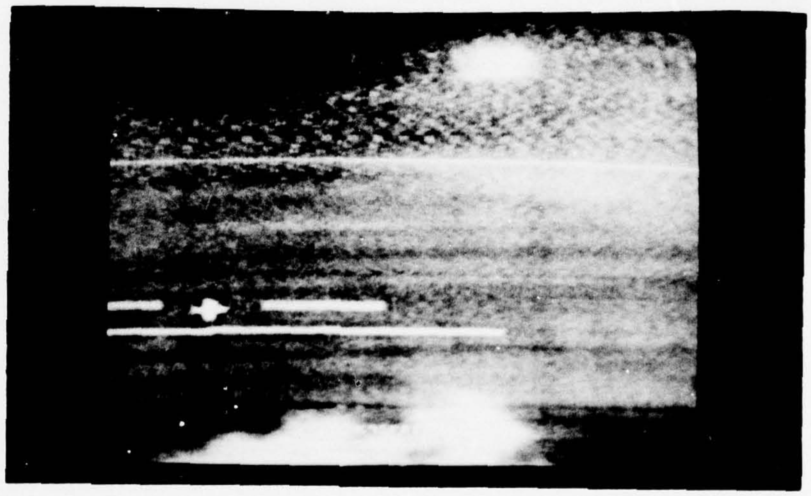


Figure 4-9 Central Portion of Frame K10SP1 (Figure 4-5) as Viewed when Displaying every Picture Element.

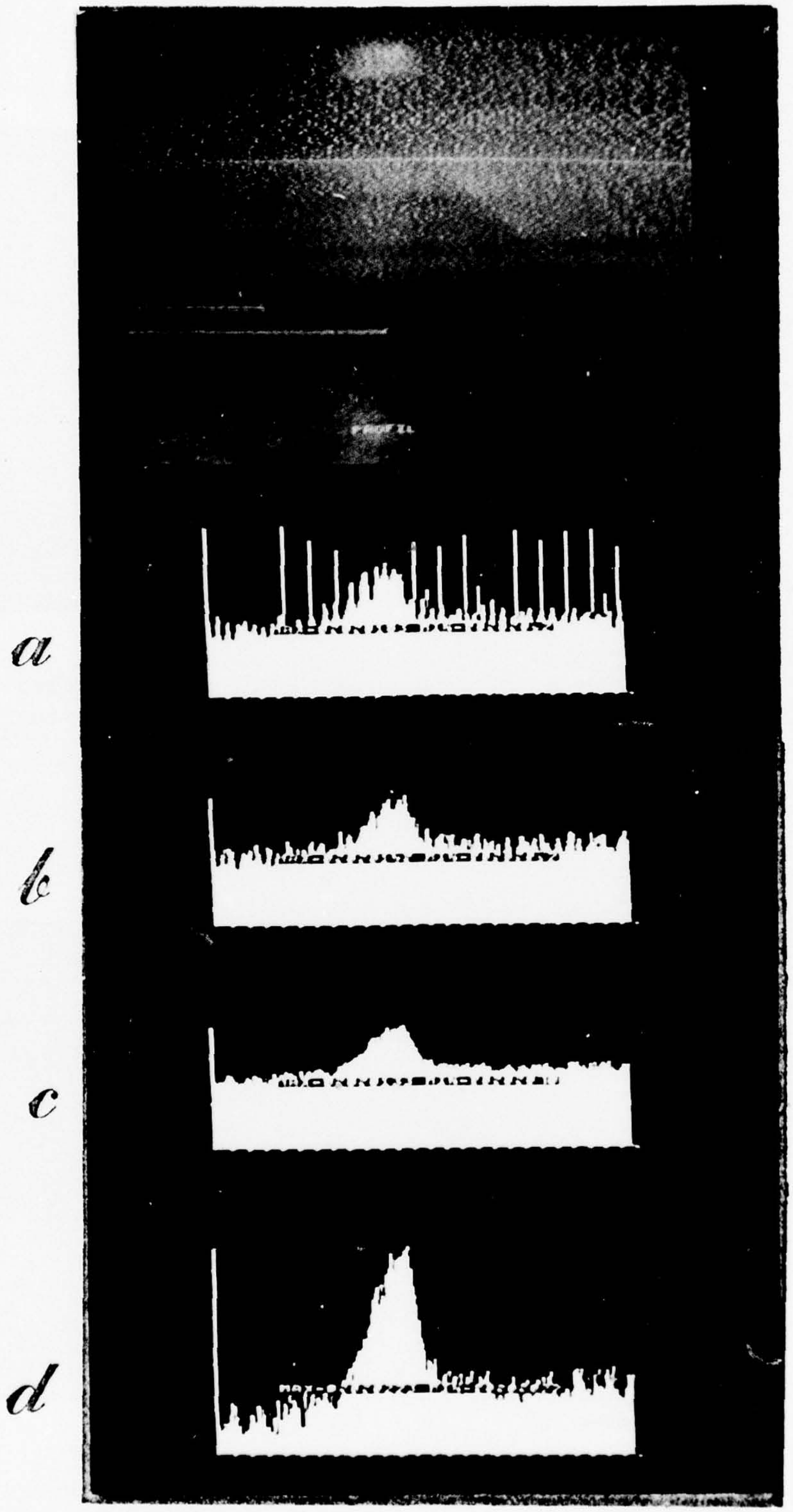


Figure 4-10 Grey Level Profiles (plot of intensity vs. position) Across the Scan Line Indicated between the Arrows in Frame K00010 are Shown for (a) K00010; (b) K100D0; (c) K10SP0; and (d) K10SP1

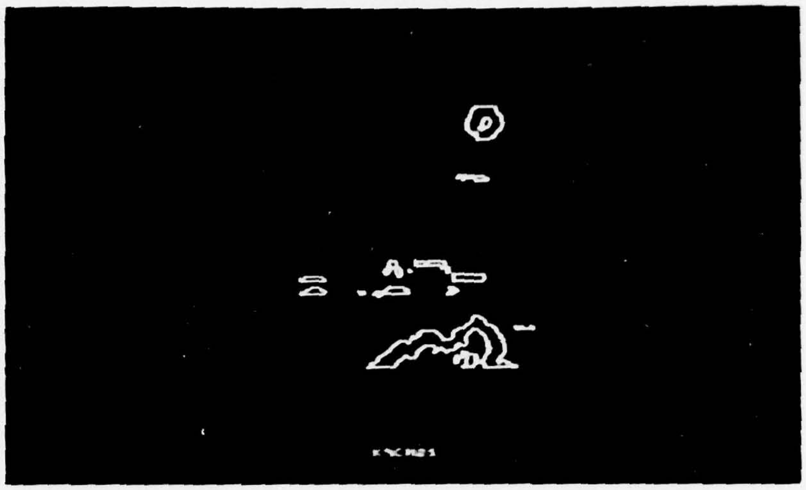


Figure 4-11 Display Showing Boundaries Extracted From Frame K00010 Prior to the Application of the Cueing Logic.

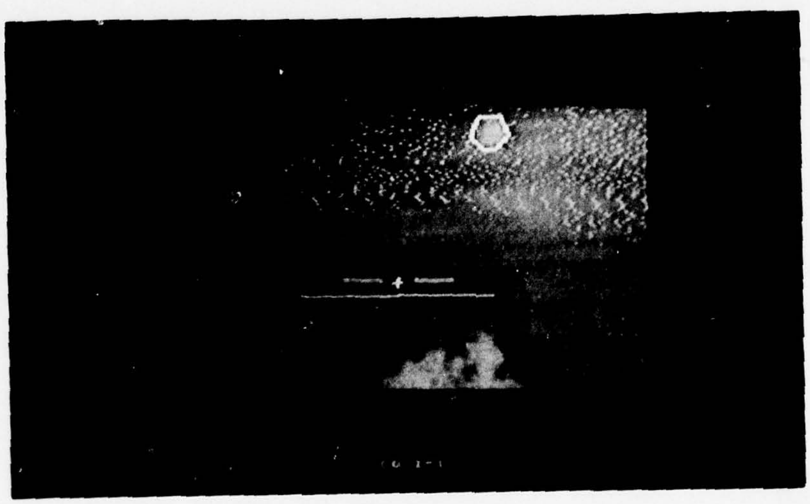


Figure 4-12 Display Showing Boundary (overlaid on Frame K00010) Remaining after Filtering with Boolean Logic Designed to Detect Tactical Target Shapes.

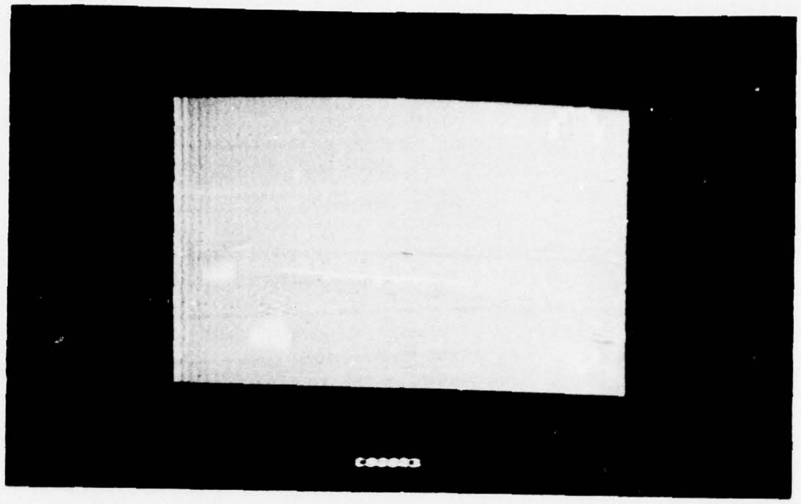
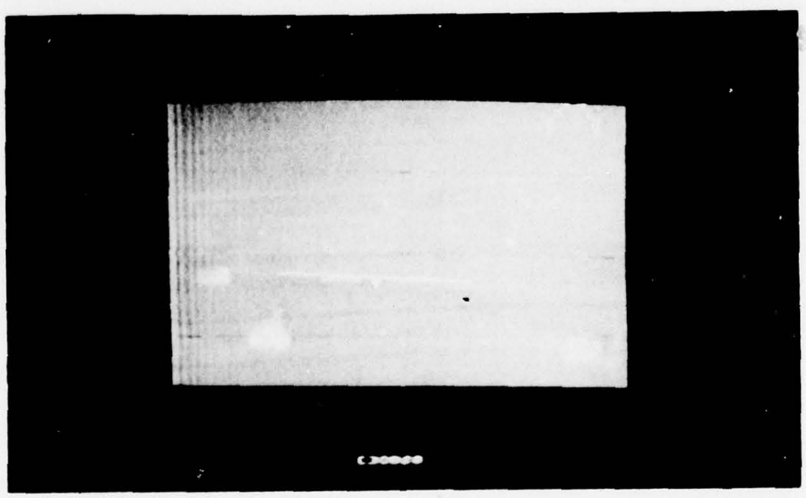


Figure 4-13 Frame C0000 4 as Displayed on DICIFER TV Monitor. The Targets in View are (from left to right) APC, Tank, and 2-1/2 Ton Truck. Displayed Image has been Reduced via Pixel Deletion.



Frame 4-14 Frame C40D00. Results of Processing Frame C00004 for Removal of Salt-and-Pepper Noise. Displayed Image has been Reduced via Pixel Deletion.

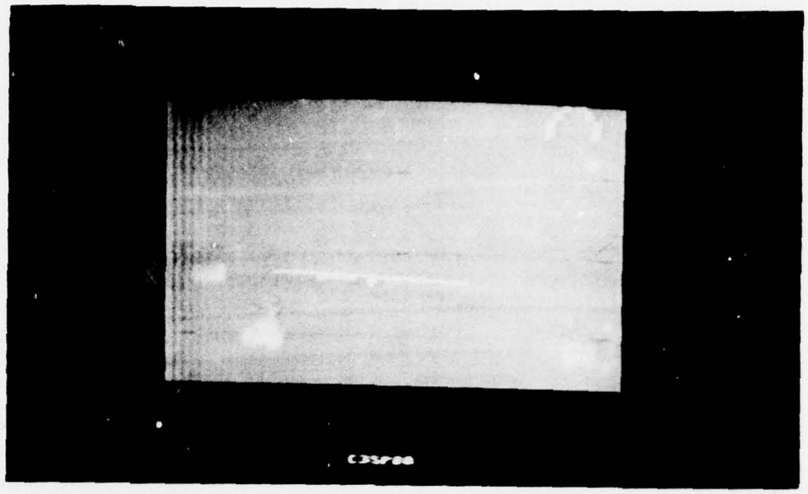


Figure 4-15 Frame C4SP00. Results of Applying Spatial Filtering to Frame C40D00. Effect is to Suppress High-Frequency "Ripple" Noise Pattern Along the Scan Line. Displayed Image has been Reduced via Pixel Deletion.

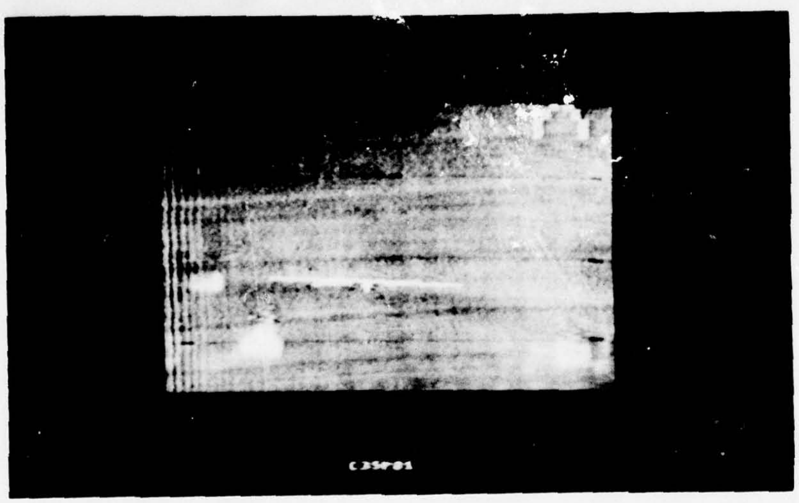


Figure 4-16 Frame C4SP01. Results of Applying Dynamic Range Change to Frame C4SP00 in Order to Achieve Contract Enhancement. Displayed Image has been Reduced via Pixel Deletion.

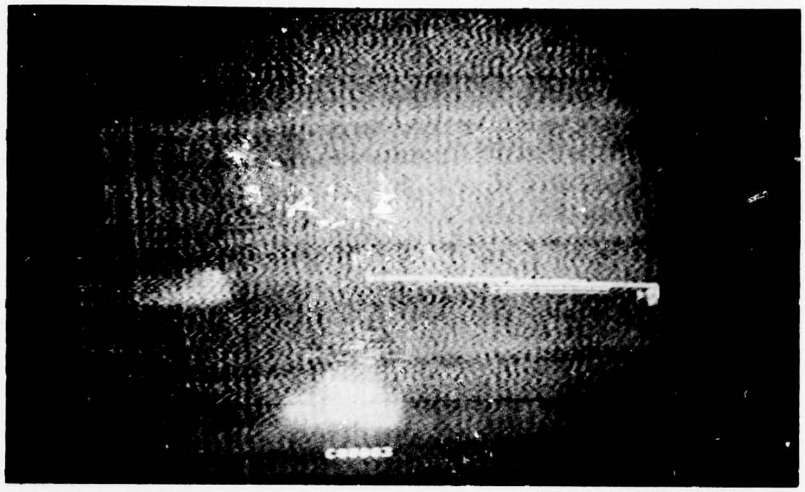


Figure 4-17 Left Portion of Frame C00004 (Figure 4-13) as Viewed when Displaying every Picture Element.

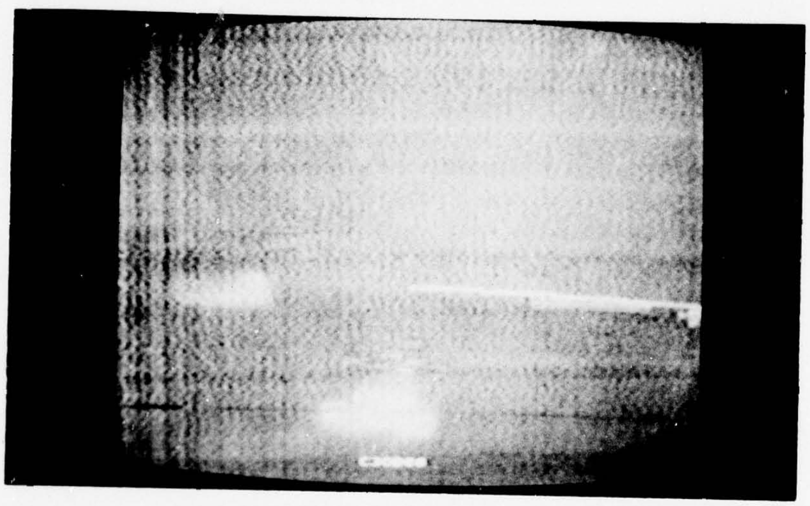


Figure 4-18 Left Portion of Frame C40D00 (Figure 4-14) as Viewed when Displaying every Picture Element.

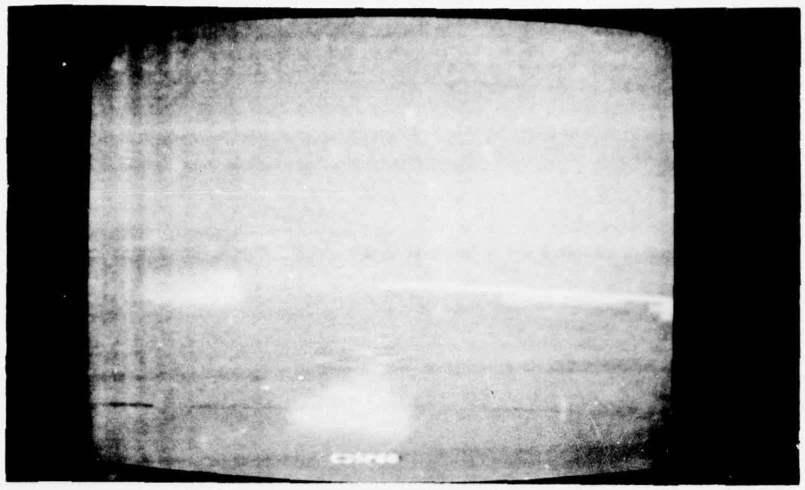


Figure 4-19 Left Portion of Frame C4SP00 (Figure 4-15) as Viewed when Displaying every Picture Element.

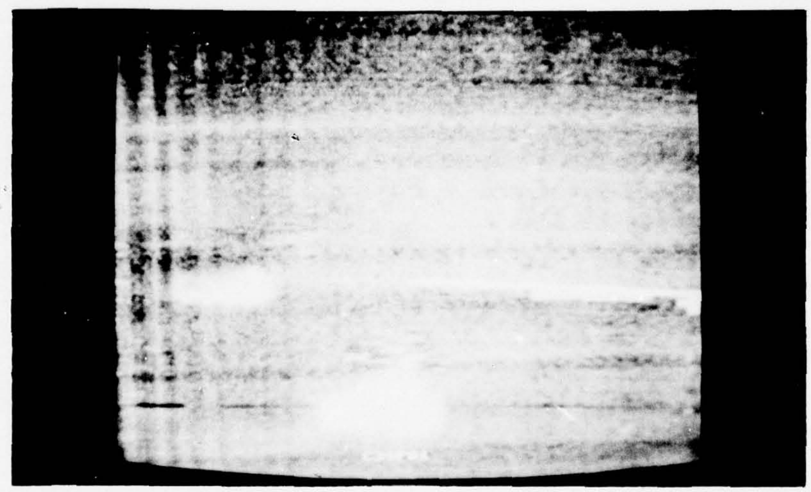


Figure 4-20 Left Portion of Frame C4SP01 (Figure 4-16) as Viewed when Displaying every Picture Element.

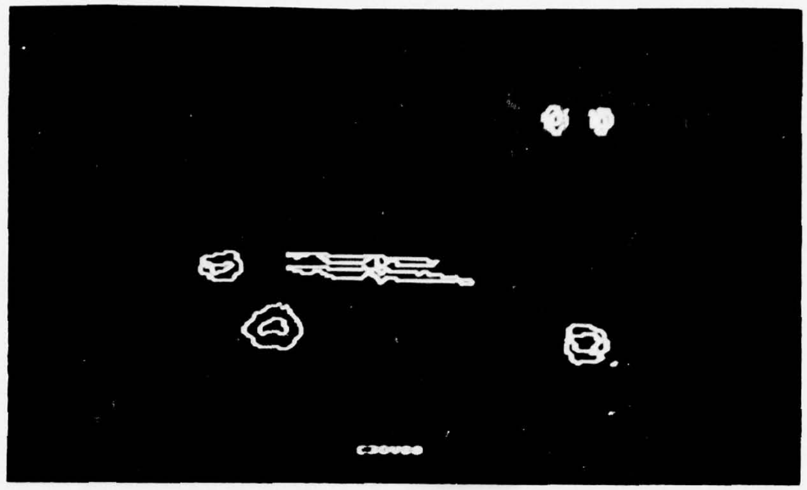


Figure 4-21 Display Showing Boundaries Extracted from Frame C00004 Prior to the Application of the Cueing Logic.

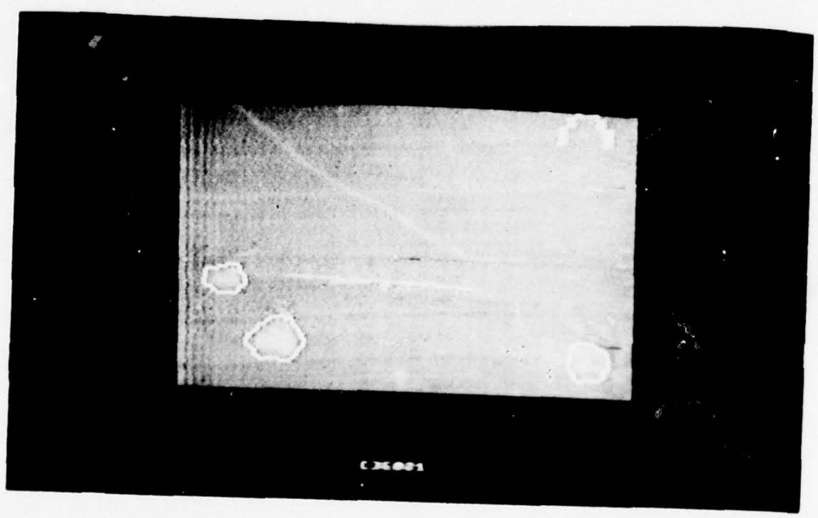


Figure 4-22 Display Showing Boundaries (overlaid on Frame C00004) Remaining after Filtering with Boolean Logic Designed to Detect Tactical Target Shapes.

4.6. EDGE DETECTION

The purpose of an edge detection routine is to determine those image points that are on object-background boundaries. In a multi-grey-level image, object boundaries provide a grey level gradient due to differences between the average brightness of an object and its background.

Two basic varieties of edge detection routines exist on the DICIFER system. The first and simplest routine available is called "Point Edge Detection" due to the fact that it operates on a point and the immediate neighbors of the point. To review the operation of the algorithm, let us represent a given pixel and its eight neighbors as follows:

1	2	3
0	X	4
7	6	5

where X represents the pixel and the numbers 0 through 7 index its eight neighbors. One or more of the neighbors must be used for calculating a gradient. The simplest gradient consists of the absolute difference between the value of pixel X and that of one of the selected neighboring points. Variants of this algorithm would calculate the sum or maximum of absolute differences for several neighbors.

The "Area Edge Detection" routine extends the Point Edge concept by considering a large number of the neighboring points. Again the number of points is controlled by the user, but in a slightly different manner. Consider the following diagram (Figure 4-23):

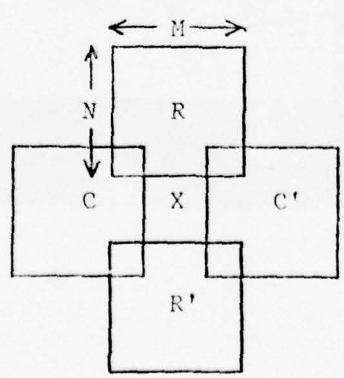


Figure 4-23 Neighborhoods Used for Gradient Calculation

R, R', C, and C' are rectangular arrays of points with specified dimensions of M by N. Each array borders on the central pixel X. For each X, the average value for each of the four arrays is calculated and the two absolute differences of these values, $|R - R'|$ and $|C - C'|$, are available to be compared to a preselected threshold for making an edge-point/non-edge-point decision. Variants of this scheme use either the maximum difference or the sum of the differences.

After the gradient has been calculated by one of the previously described methods, both edge detection routines simply check the gradient(s) and compare them with a preselected threshold value. If the gradient exceeds the threshold, the corresponding pixel is labelled "1" (edge point), otherwise "0" (non-edge point). The resulting binary image is then available for analysis.

4.6.1. Considerations for the Use of Edge Detection Algorithms

The Point Edge Detection algorithm suffers from two problems due to its method of calculating gradients.

- o Gradient calculations based on only a few neighboring pixels are often sensitive to image noise. A noise point could deviate significantly from its neighbors and would therefore produce a detectable grey level difference. Depending upon the magnitude of this difference relative to the selected threshold, a false edge may be indicated. Should this occur frequently, the remainder of the object identification process would be greatly complicated due to the false edges.

- o A second undesirable feature of gradient calculation based on only a few neighboring pixels is its insensitivity to a "broad" edge, i.e., an edge that is several pixels wide. In this case, since the Point Edge routine is restricted to the immediate neighborhood of the point, a distinct contrast between an object and its background is completely overlooked. Instead, the gradient across the edge is measured which may be very low and which may not exceed the selected threshold. Indeed, it may be so low at some points that it is well below the noise level, making it impossible to detect.

In comparison, the Area Edge Detection algorithm is less sensitive to noise because of the averaging effect and more sensitive to broad edges because of the larger area considered. This is especially true concerning the FLIR data, which has a very noisy backgrounds and large "fuzzy" targets. Therefore, the Area Edge Detection routine was chosen as the best means available on DICIFER for object detection.

4.7. APPROACH FOR FLIR TARGET DETECTION

The sequence of DICIFER routines that has been used to perform the target cueing, starting from the input of a preprocessed image, is depicted in Figure 4-24.

The FLIR data were found to be oversampled in terms of inherent resolution. Therefore, a decimation of the image was performed prior to use of the Area Edge Detection. By keeping only every fourth row and every fourth column, a decimated image was produced which was not significantly degraded in image information content. This decimated image was much easier to work with, both in terms of storage space and processing time, and may have actually helped decrease the amount of background noise picked up by the Area Edge Detection.

4.7.1. Application of "Area Edge Detection"

The output binary images from an Area Edge Detection performed on the decimated images K9SP01 and C3SP01 are shown in Figures 4-25 and 4-26, respectively. As can be seen in these output images, Area Edge Detection produces a thick boundary, usually consisting of several pixels. The advantage to this type of boundary is that the boundary tends to remain complete around the object. This is an important aspect when features based on the spatial extent of the object boundaries are extracted. A gap in a boundary would cause features such as enclosed area and perimeter to become completely distorted.

An important variable in Area Edge Detection is the threshold level. The threshold value is an input parameter to the Area Edge Detection routine and must be provided by the user. The value chosen will greatly affect the output image and all following results. The aim of this processing was to create a well-defined, unbroken boundary around all objects while reducing background noise to a minimum. The effective use of this technique requires that parameters be selected which will provide for the extraction of unbroken objects around the tactical target shapes. Excessive image noise or contrast variations near an object edge could cause gaps in the object boundaries which would result in the calculation of uncharacteristic (for tactical targets) measurements of shape and size. It may be possible to consider techniques for filling in such gaps based on a priori knowledge of target signatures. However, since such algorithms do not presently exist on DICIFER, such boundary extension techniques remain as topics for future investigations.

Given the above considerations, a threshold value was chosen to meet these requirements. That is, after experimentation and analysis, a threshold value of 50 grey values was chosen to go with a box size given

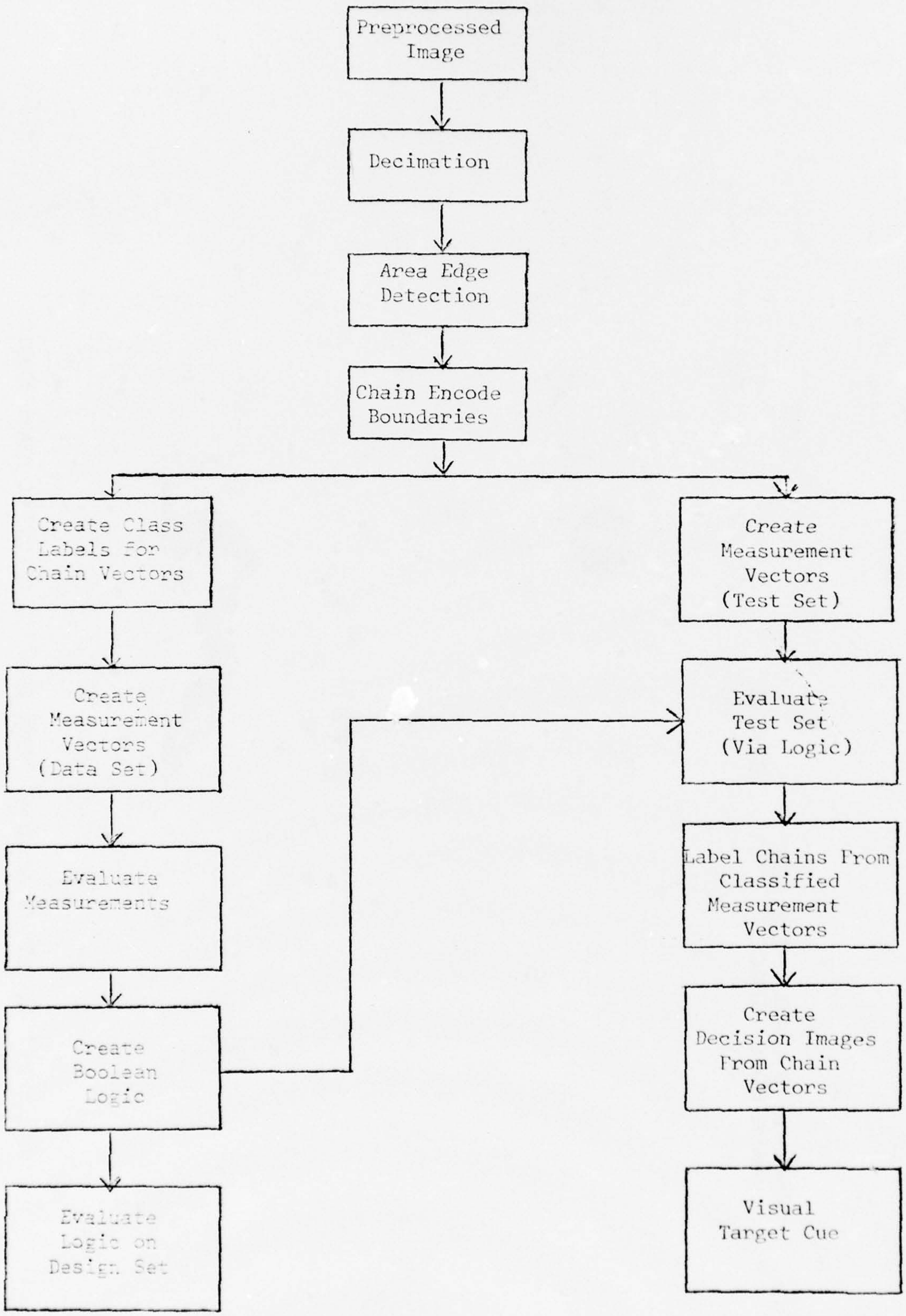


FIGURE 4-24 TARGET DETECTION PROCESSING FLOW

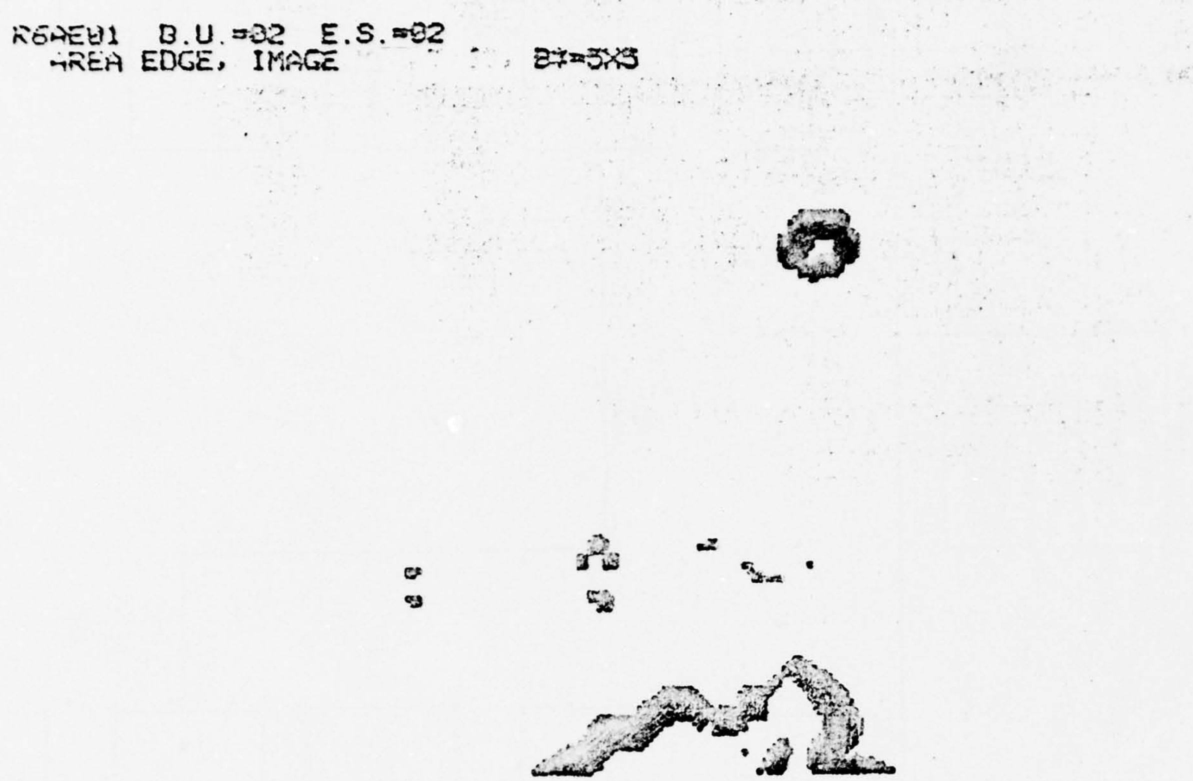


FIGURE 4-25 RESULTS OF APPLYING "AREA EDGE DETECTION" TO FRAME K00010

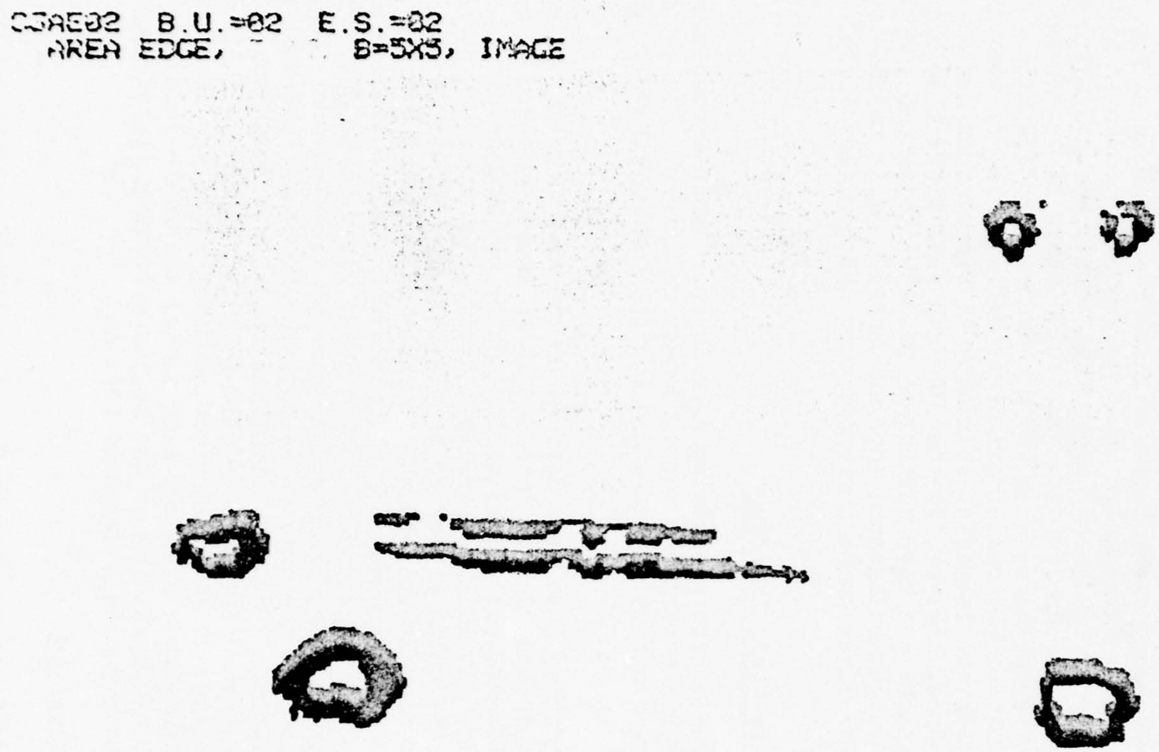


FIGURE 4-26 RESULTS OF APPLYING "AREA EDGE DETECTION" TO FRAME C00004

by $M = N = 5$ (as shown in Figure 4-23). If the maximum of $|R-R'|$ and $|C-C'|$ exceeded the threshold value (50), then the subject pixel was taken to be an edge point. Otherwise, the subject pixel was labelled as a non-edge point. Notice that for large targets this technique produces boundary edges which are several pixels thick resulting in an apparent "doughnut" shape being extracted. These values have proven to be the optimal values over the range of images that were processed. The reason that this one value was applicable over a wide range of images was that the final image preprocessing step was a contrast normalization. The contrast normalization had the effect of stabilizing the optimal gradient threshold value between images of initially different contrast. A negative consequence, however, is that low contrast targets in high contrast scenes might not have sufficient gradient to exceed the threshold, unless the target-background edge was sharp. It is recommended that contrast normalization be performed over smaller (localized) portions of the image.

4.7.2. Boundary Chain Encoding

The next step in the processing flow is the "Chain Encode Boundaries" routine. This routine performs two different functions: (1) the detection and listing of binary objects, and (2) the extraction of certain object features.

Object boundaries have been defined by the Area Edge Detection routine but that information is imbedded in an image format. In an image (raster) format, each pixel is indexed by row and column, i.e., its position only. It is necessary to identify each boundary pixel as belonging to a particular object to be classified. "Chain Encode Boundaries" is a process whereby the sequence of adjacent pixels along an object boundary is encoded as a chain vector. Each chain vector then describes one and only one object in the image field. The data for each chain consists of a header block and a chain of link vectors. Each link vector connects adjacent points on an object's boundary, and must take one of eight directions, as each pixel has only eight neighbors. The unit vectors are then ordered consecutively with respect to their position on the boundary. The header block contains ancillary data and features extracted from the object edge during the chain encoding. Appendix A gives a description of the boundary tracing algorithm employed by the Chain Encode Boundaries routine.

4.8. FEATURE EXTRACTION

Object classification schemes depend on object features which can be compared by various logical systems. Therefore, classification accuracy is limited by the features provided to the logic. With the FLIR data, object size and shape, calculable from the previously generated object boundaries, were the features upon which the classification was based.

The Chain Encode Boundaries routine discussed previously extracts several features from each object boundary. The shape information is encoded by the chain of unit direction links, which describe the trace of the boundary points around the object. This shape information, although a complete description of the shape, is not in a usable form; however, in addition to the chain links, the following spatial features are extracted from the boundary: (1) enclosed area, (2) perimeter, (3) minimum and maximum row and column coordinates of the boundary, (4) an interior edge boundary flag, and (5) an edge-limited vector flag.

A sample listing of the header information in the chain vector file, C4CH00, is shown in Table 4-2. File C4CH00 is the result of applying the preprocessing, edge detection, and chain encoding processes to frame C00004.

4.9. OBJECT CLASSIFICATION

Once objects have been detected and features extracted, one of many well-developed techniques for classification can be used. OLPARS, a subset of the DICIFER system, has an extensive repertory of routines to create, test, analyze, and utilize various classification logic schemes. It was decided that Boolean logic would be the most effective type of logic for this problem. This decision was based on two considerations: (1) that the simple hypercube logic would be sufficient for the types of features and classes found in the FLIR images, and (2) that a Boolean logic classifier could be simply and cheaply implemented in hardware and thus would be suitable as part of a FLIR system.

4.9.1. Logic Design

The left side of the process flow in Figure 4-24 represents the logic design. The first step is the selection of a design or training set. This design set must be carefully chosen so that all class types are adequately represented. The design set chosen consisted of four chain vector files containing a total of four targets. The four images represented in this design set were images K4, L4, L5, and L6*, and the corresponding chain vector files are shown in Figures 4-27, 4-28, 4-29, and 4-30, respectively.

The first software process involves creating the various classes by labelling the chain vectors. This is accomplished by the user entering a class symbol for each vector in the design set as it is individually

---

* As a convenient shorthand, the image from which a processed image originated will be identified by a letter representing the tape and a number representing the individual image frame on that tape. Thus L5 is the 5th image frame from tape L.

EDGE LIM.	IN OUT	X	Y	AREA	DIAG CNT	ORTH CNT	PERIM	P*P/A	WIDTH	HEIGHT
0	0	174	5	74.	15.	18.	39.	21.	12	11
0	0	197	5	44.	15.	32.	53.	64.	10	10
0	1	176	8	16.	5.	8.	15.	14.	4	5
0	0	22	70	180.	13.	36.	54.	16.	18	13
0	0	50	71	152.	22.	124.	155.	158.	70	6
0	1	13	76	4.	4.	2.	8.	15.	3	2
0	1	21	76	26.	8.	12.	23.	21.	9	5
0	0	50	77	261.	31.	146.	190.	138.	87	7
0	0	46	94	376.	30.	36.	78.	16.	26	21
0	1	42	102	62.	10.	20.	34.	19.	12	7
0	0	185	102	272.	19.	40.	67.	16.	19	18
0	1	189	107	87.	13.	24.	42.	21.	14	9
0	0	205	119	0.	1.	2.	3.	255.	1	1

Table 4-2. Header Information for Chain Vector File C4CH00

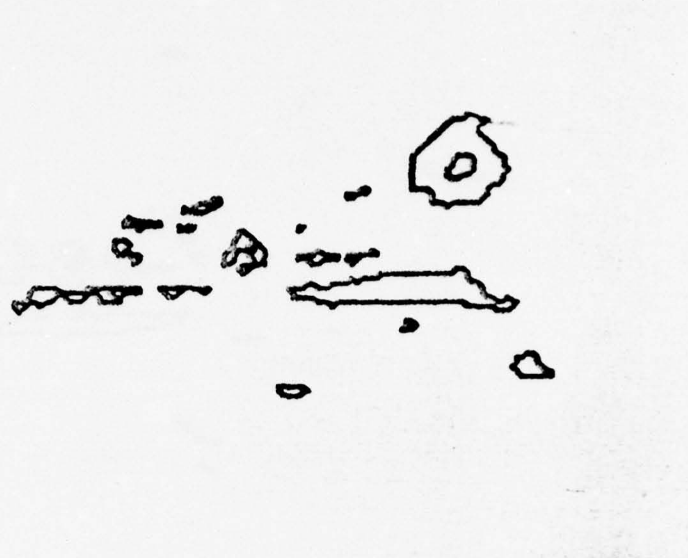


FIGURE 4-27 DISPLAY OF CHAIN VECTORS FOR FRAME K4

FIGURE 4-28 DISPLAY OF CHAIN VECTORS FOR FRAME L4

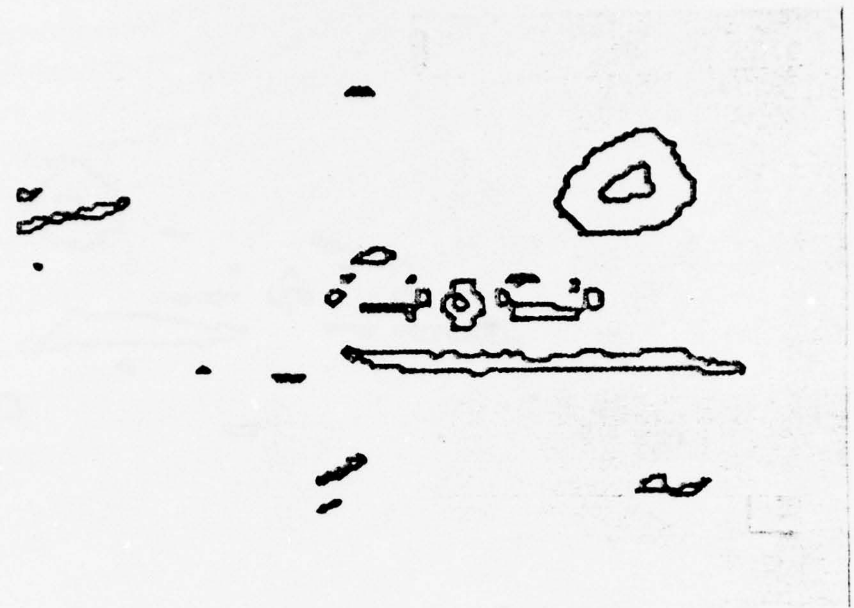


FIGURE 4-29 DISPLAY OF CHAIN VECTORS FOR FRAME L5

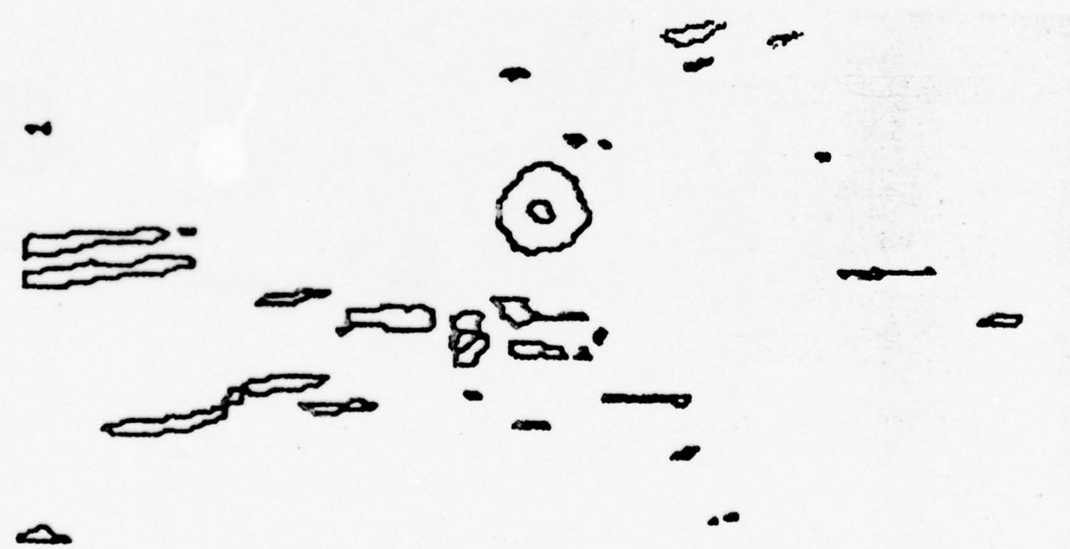


FIGURE 4-30 DISPLAY OF CHAIN VECTORS FOR FRAME L6

displayed on the terminal. Although only a target vs. non-target classification was necessary, it was logically more convenient to create five different classes. The five classes created were targets, noise, markers, inside edges, and others; represented respectively by the symbols T, N, M, I, and O.

The target class obviously represents the tactical targets and the noise class represents small "objects" caused by background noise. The marker class represents the graphic overlay of a cross with extending lines placed near the middle of the image field. The inside edge class identifies those chains which are the interior edges of the thick boundaries produced by the Area Edge Detection. The "other" category represents all objects not identified as one of the previous four classes.

The next processing step was to put the information contained in the chain vectors into a format which could be subsequently processed by the logic routines. This format is a measurement vector; the following seven measures were taken (or formed) from the features in the chain vector: (1) Area, (2) Perimeter, (3) $(\text{Perimeter})^2/\text{Area}$, (4) Width, (5) Height, (6) Edge-Limited Marker, and (7) Inside Edge Marker.

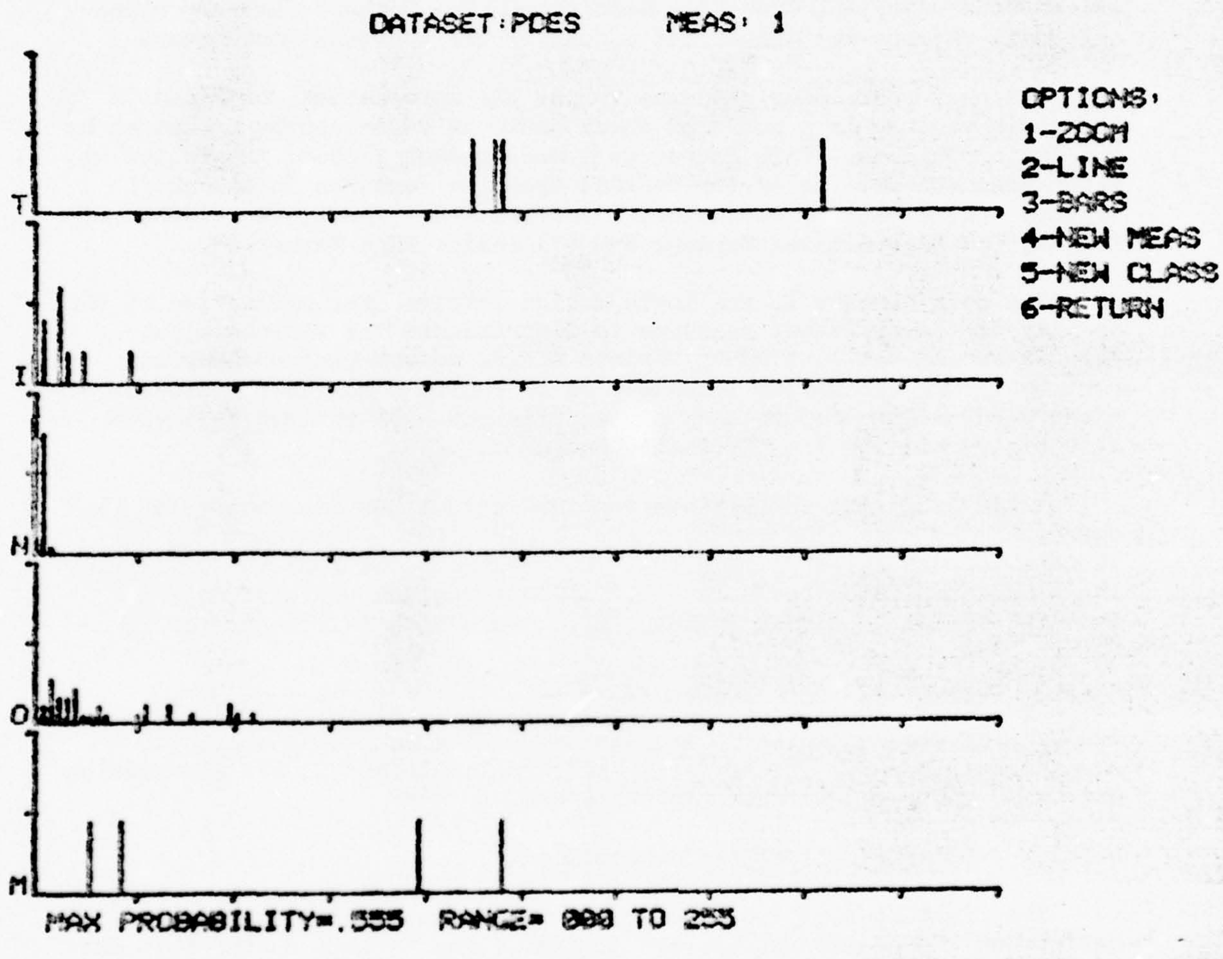
The next process in the logic design involves the evaluation of the ability of the different features to discriminate the various classes. This is easily facilitated by various OLPARS measurement evaluation routines. One evaluation technique is to employ a multiple histogram of measurement magnitude for each class. Figures 4-31 through 4-35 show such histograms for the first five measures.

After analyzing these histograms, a logical Boolean expression was developed for each class, except the "other" class which contains all vectors not otherwise classified. These expressions form the classifier logic and are listed in Table 4-3. The expressions are then entered into the system to form a Boolean logic tree which is illustrated in Figure 4-36.

The final step in the logic design is the evaluation of logic against the data on which it was designed. The logically assigned classes are compared to the design set labelled classes, and a confusion matrix is generated as shown in Table 4-4.

4.9.2. Classification Logic Evaluation

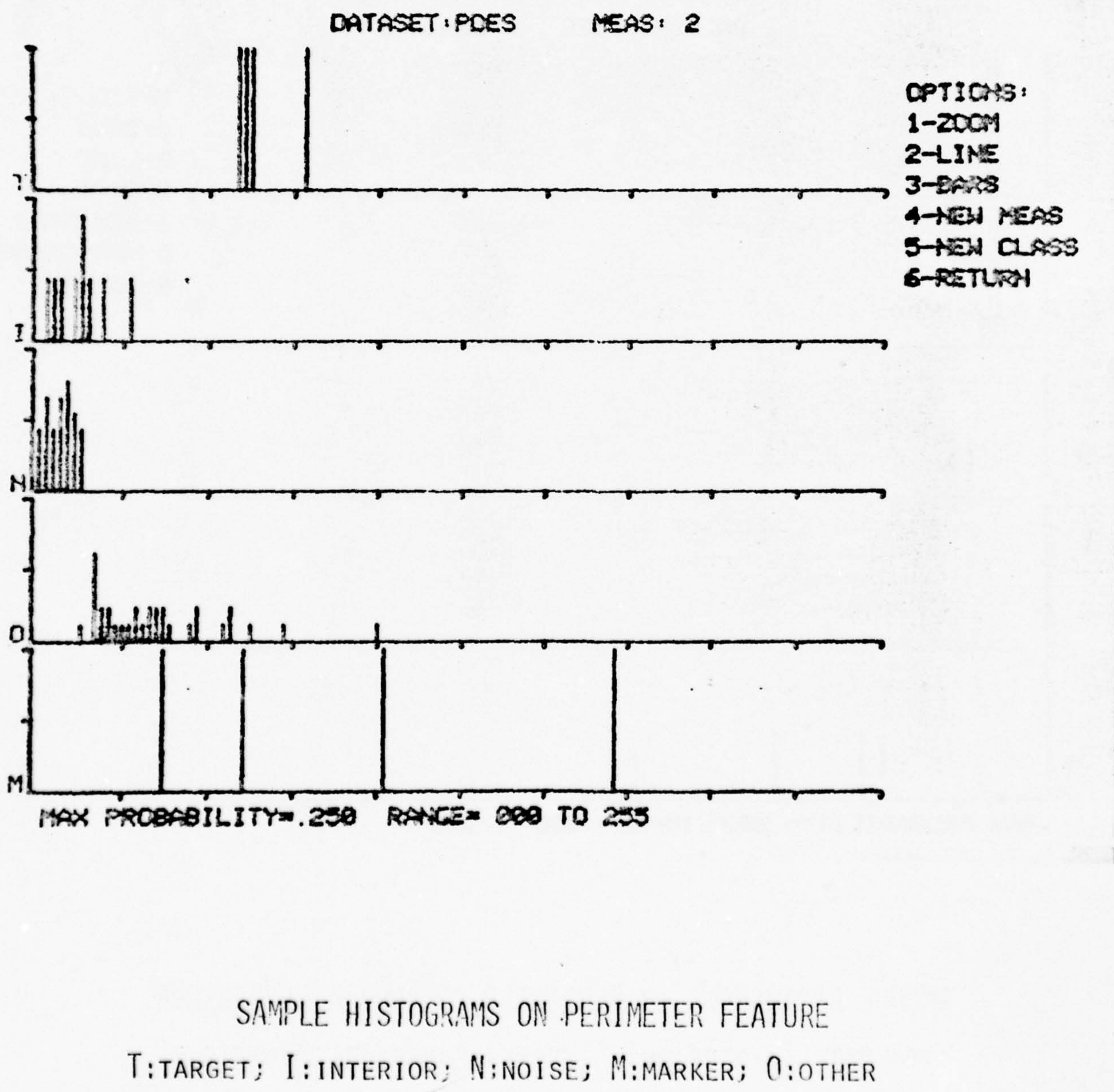
The right side of the process flow shown in Figure 4-24 consists of performing the classification and cueing of the targets on the FLIR data set. The first step is to create measurement vectors as in the logic



SAMPLE HISTOGRAMS ON AREA FEATURE

T:TARGET; I:INTERIOR; N:NOISE; M:MARKER; O:OTHER

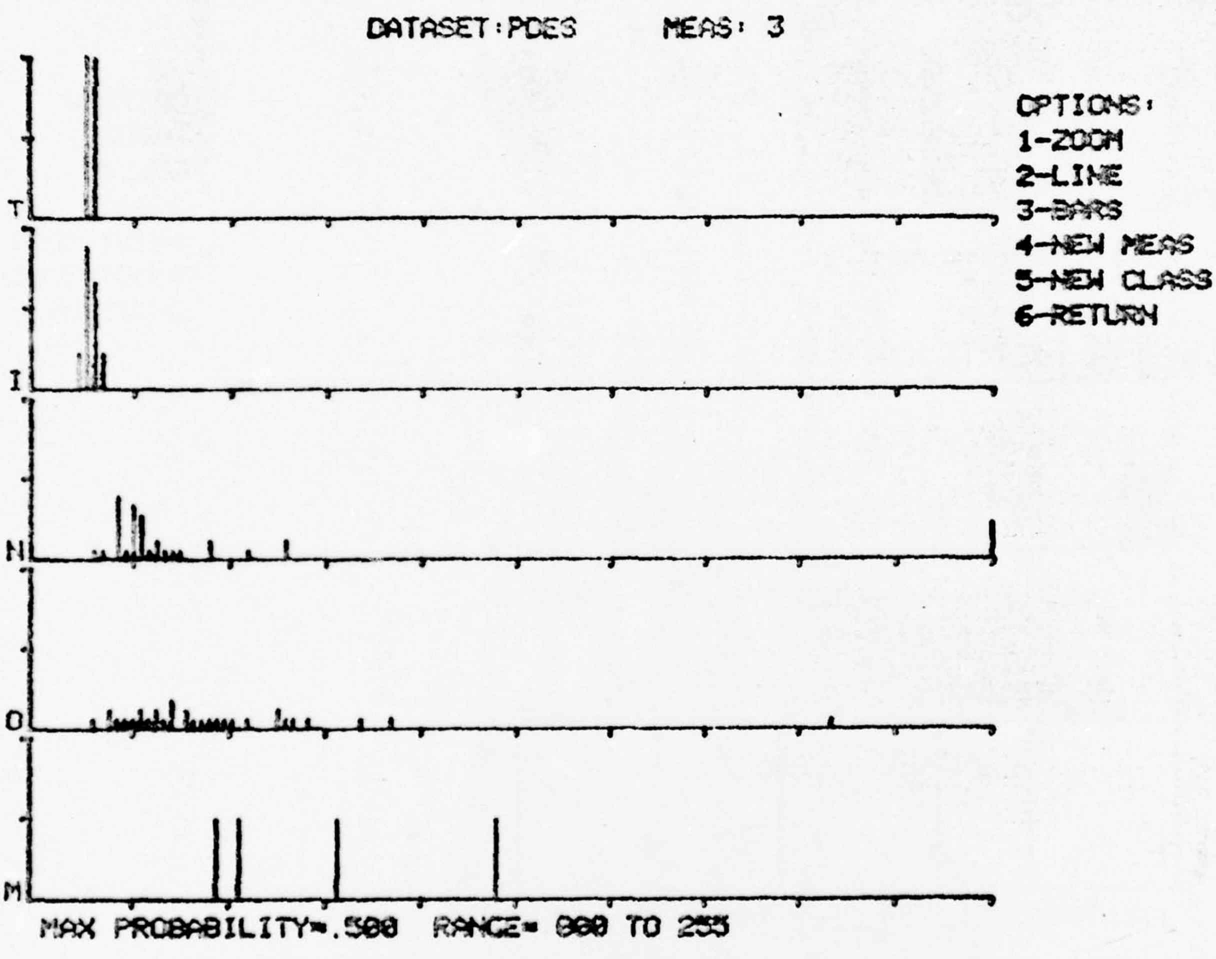
FIGURE 4-31



SAMPLE HISTOGRAMS ON PERIMETER FEATURE

T:TARGET; I:INTERIOR; N:NOISE; M:MARKER; O:OTHER

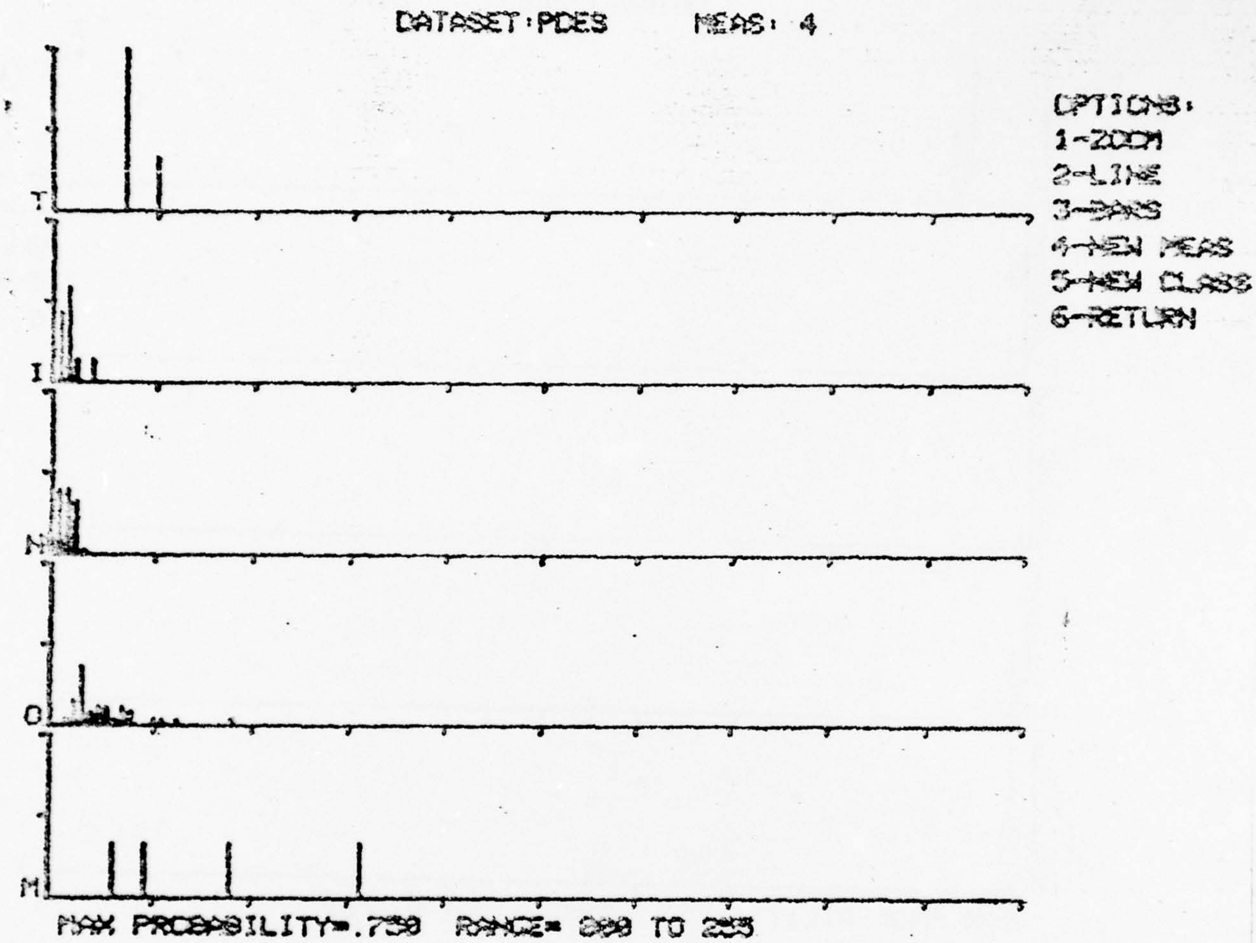
FIGURE 4-32



SAMPLE HISTOGRAMS ON PERIMETER SQUARED/AREA FEATURE

T:TARGET; I:INTERIOR; N:NOISE; M:MARKER; O:OTHER

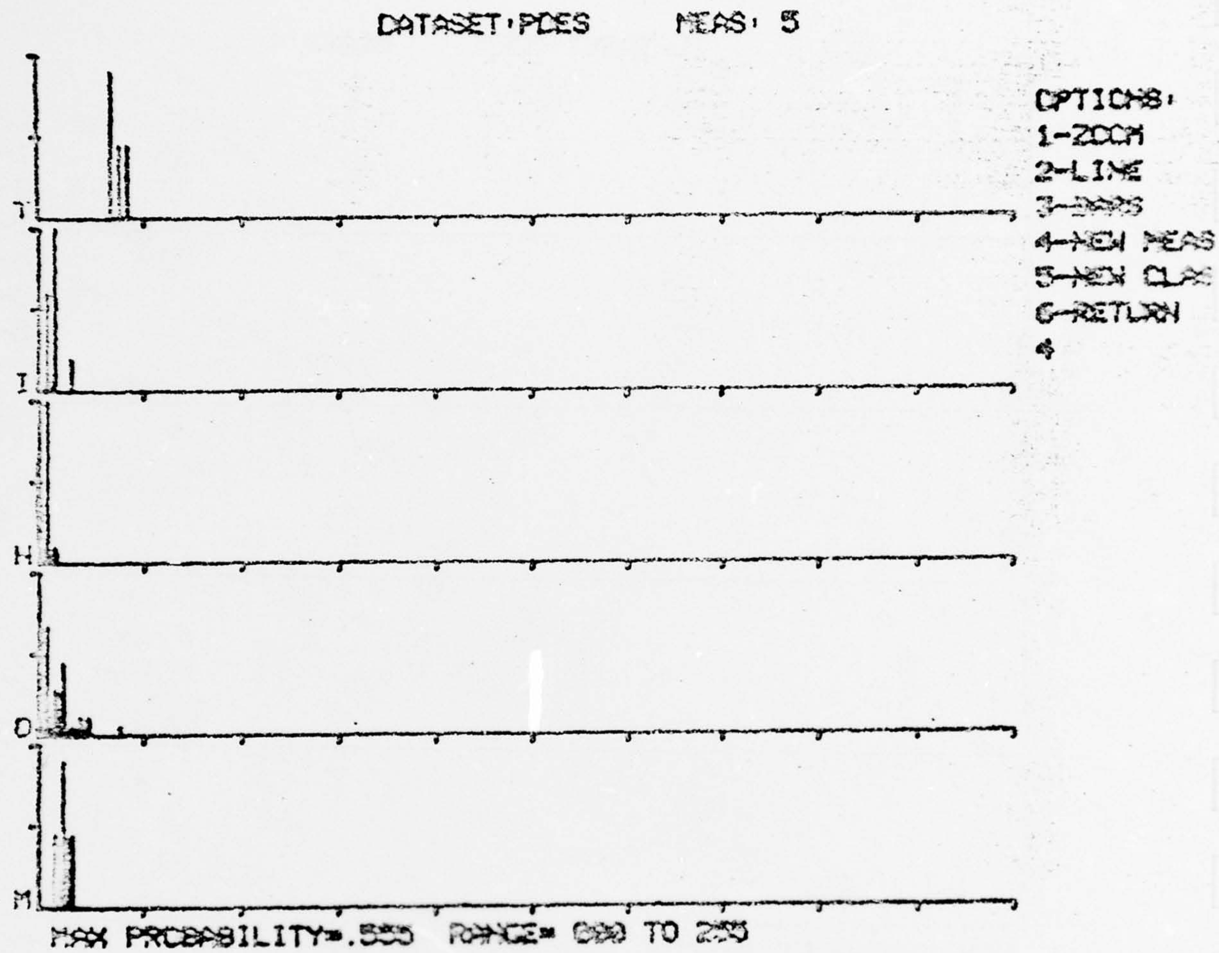
FIGURE 4-33



SAMPLE HISTOGRAMS OF WIDTH FEATURE

T:TARGET; I:INTERIOR; N:NOISE; M:MARKER; O:OTHER

FIGURE 4-34



SAMPLE HISTOGRAMS ON HEIGHT FEATURE

T:TARGET; I:INTERIOR; N:NOISE; M:MARKER; O:OTHER

FIGURE 4-35

CLASSIFIER LOGIC (BOOLEAN)

NODE Ø:

$M7 > \emptyset$

NODE 2:

$(M1 \Leftarrow 5) \wedge (M2 \Leftarrow 15)$

NODE 4:

$(M1 > 1\emptyset) \wedge (M2 > 35) \wedge (M3 > 4\emptyset) \wedge (M4 > 15) \wedge (M5 < 15)$

NODE 6:

$(M1 > 75) \wedge (M2 > 4\emptyset) \wedge (M2 < 125) \wedge (M3 < 3\emptyset)$

FEATURES:

M1: AREA

M2: PERIMETER

M3: PERIMETER SQUARED OVER AREA $(M2)^2/(M1)$

M4: WIDTH

M5: HEIGHT

M7: INTERIOR TRACE FLAG

FLIR TARGET DETECTION LOGIC

(ANVL DATA: LARGE TARGETS)

Table 4-3. Description of Boolean Logic

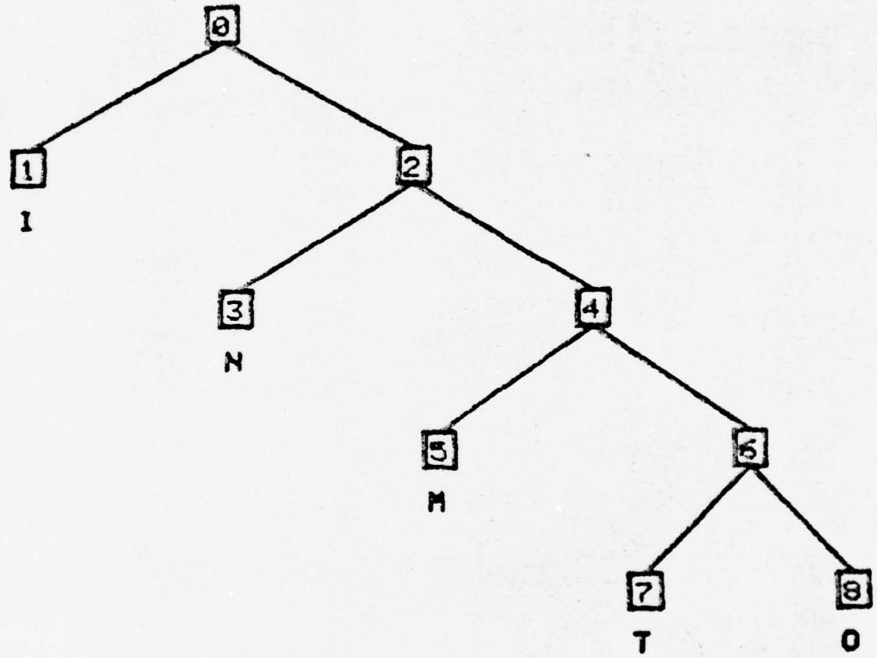


FIGURE 4-36 STRUCTURE OF BOOLEAN LOGIC

Labelled As \ Classified As	LOGIC FILE PDES			VECTOR FILE PDES	
	I	N	M	T	O
I	9	0	0	0	0
N	0	36	0	0	0
M	0	9	0	0	0
T	0	0	0	4	0
O	0	1	6	0	25

Table 4-4. Confusion Matrix for Design Logic

design, but from unlabelled chain vectors. These measurement vectors constitute the test set. This test set is then classified by the Boolean logic classifier previously developed. A routine is then used to label the chain vectors in accordance with the corresponding measurement vector node assignments. A sample header dump of the classified chain vector C4 is listed in Table 4-5. Another routine was used to generate an image depicting edges which had been classified into a given category, as an aid in assessing the value of the classification logic.

4.9.3. Visual Cueing

Visual cueing was accomplished by overlaying the boundary images of the targets and the original image on a television display. This display creates a highly visible outline around the targets, as illustrated in Figure 4-22 on image C4. The use of the routine Binary Flicker, which flickers the binary overlay on and off at a user-prescribed rate, produces a visually striking cueing scheme.

4.10. CLASSIFICATION RESULTS

The result of the test set classification was the correct identification for 38 of 43 tactical targets in 34 different images, with one non-target misclassified as a target. Figures 4-37 (a-f) show the decision images generated for each class in frame K10. Figures 4-38 (a-c) show the decision images generated from frame C4 for all classes, target class and other class, respectively. These decision images are the clearest way of illustrating the results of the target cueing results. A tabular listing of the results by individual image frame is provided in Table 4-6.

4.11. NONCLASSIFIED TARGETS AND FALSE ALARMS

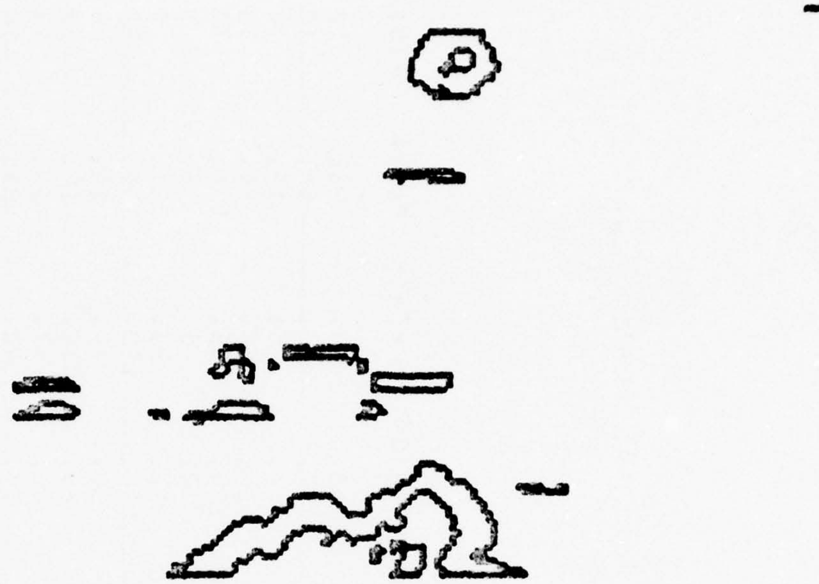
The classification scheme extended well to new data. However, an examination of the incorrectly classified targets can illuminate those processes of this scheme most subject to problems. Figures 4-39 through 4-43 show decision images of the missed targets. The top of each figure shows the boundaries of all classes in the image frame, while the bottom of each figure shows the "other" class (overlayed on the image) into which all missed target boundaries fell. Three of the five missed targets were due to overlap between the center marker graphic and the target. The overlap causes an object boundary to be defined which is representative of neither markers nor targets. The three image frames were K6, H8, and E5 and are shown in Figures 4-39, 4-40, and 4-43, respectively.

The other two missed targets were also due to other objects combining with the target to form a non-target type boundary. In the case of image frame E7 (Figure 4-41) the target was very close to a tree and the edge detection routine formed an overlapping boundary. In image frame E6 (Figure 4-42) a strong, broad noise line overlapped the target and caused a non-target boundary.

ID	EDGE LIM,	IN OUT	X	Y	AREA	DIAG CNT	ORTH CNT	PERIM	P*P/A	WIDTH	HEIGHT
O	0	0	174	5	74.	15.	18.	39.	21.	12	11
O	0	0	197	5	44.	15.	32.	53.	64.	10	10
I	0	1	176	8	16.	5.	8.	15.	14.	4	5
T	0	0	22	70	180.	13.	36.	54.	16.	18	13
M	0	0	50	71	152.	22.	124.	155.	158.	73	6
I	0	1	13	76	4.	4.	2.	8.	15.	3	2
I	0	1	21	76	26.	8.	12.	23.	21.	9	5
M	0	0	50	77	261.	31.	146.	190.	138.	87	7
T	0	0	46	94	376.	30.	36.	78.	16.	26	21
I	0	1	42	102	62.	10.	20.	34.	19.	12	7
T	0	0	185	102	272.	19.	40.	67.	16.	19	18
I	0	1	169	107	87.	13.	24.	42.	21.	14	9
N	0	0	205	119	0.	1.	2.	3.	255.	1	1

Table 4-5. Header Information for Chain Vector File C4CH00 with Classification Label (Column ID).

RSDI0A B.U.=02 E.S.=02
DECISION IMAGE, ALL CLASSES, IMAGE



RSDI0T B.U.=02 E.S.=02
DECISION IMAGE, CLASS T, IMAGE

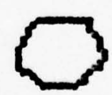


FIGURE 4-37 DECISION IMAGES FOR IMAGE K10;(a) TOP, (b) BOTTOM

R6D10I B.U.=02 E.S.=02
DECISION IMAGE, CLASS I, IMAGE

R6D10N B.U.=02 E.S.=02
DECISION IMAGE, CLASS N, IMAGE

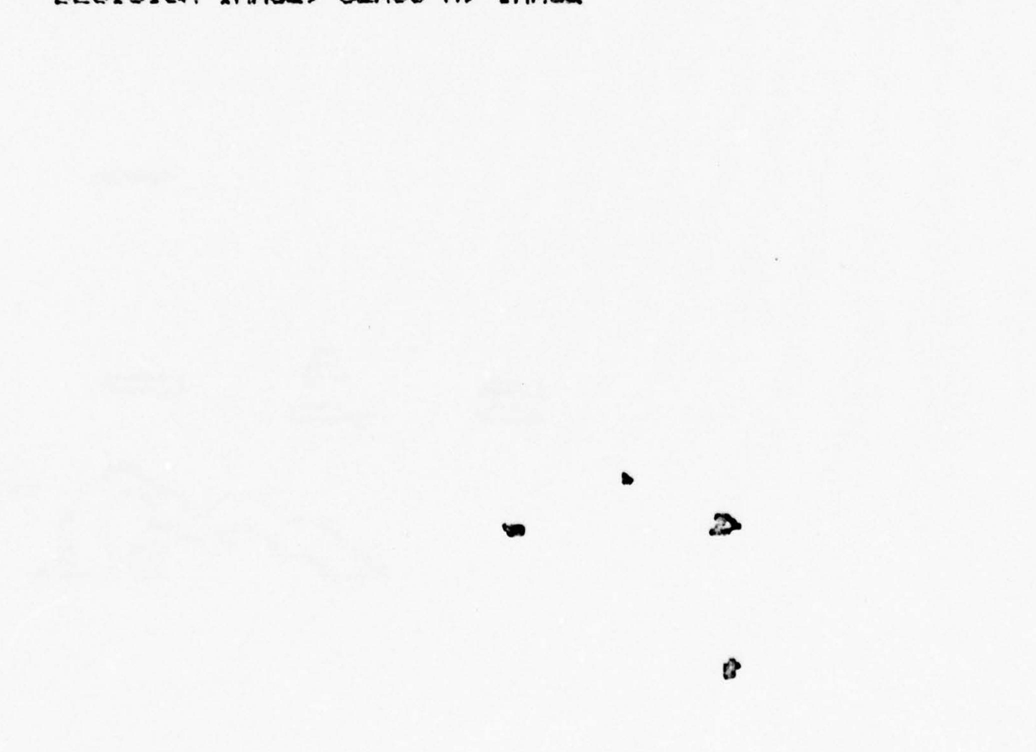
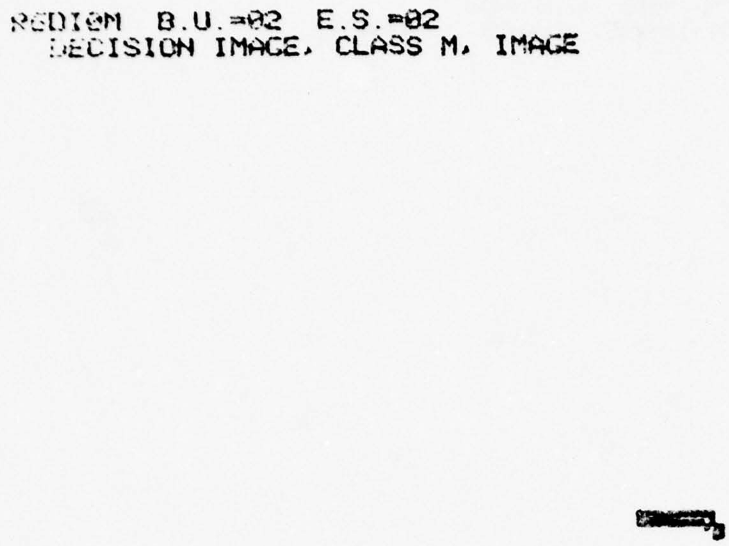


FIGURE 4-37; (c) TOP, (d) BOTTOM



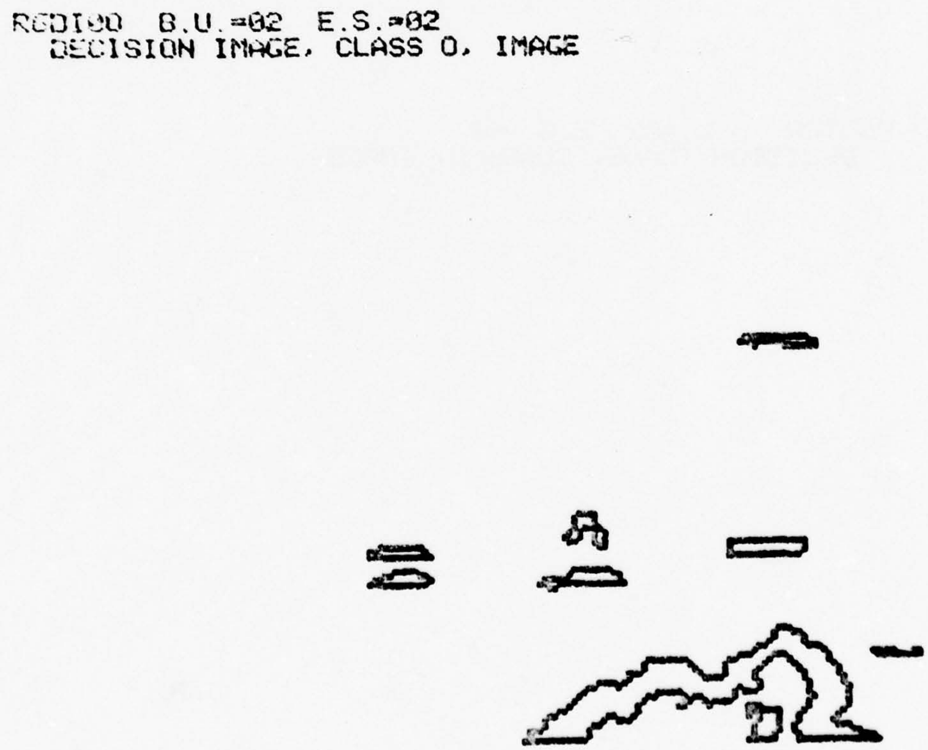


FIGURE 4-37; (e) TOP, (f) BOTTOM

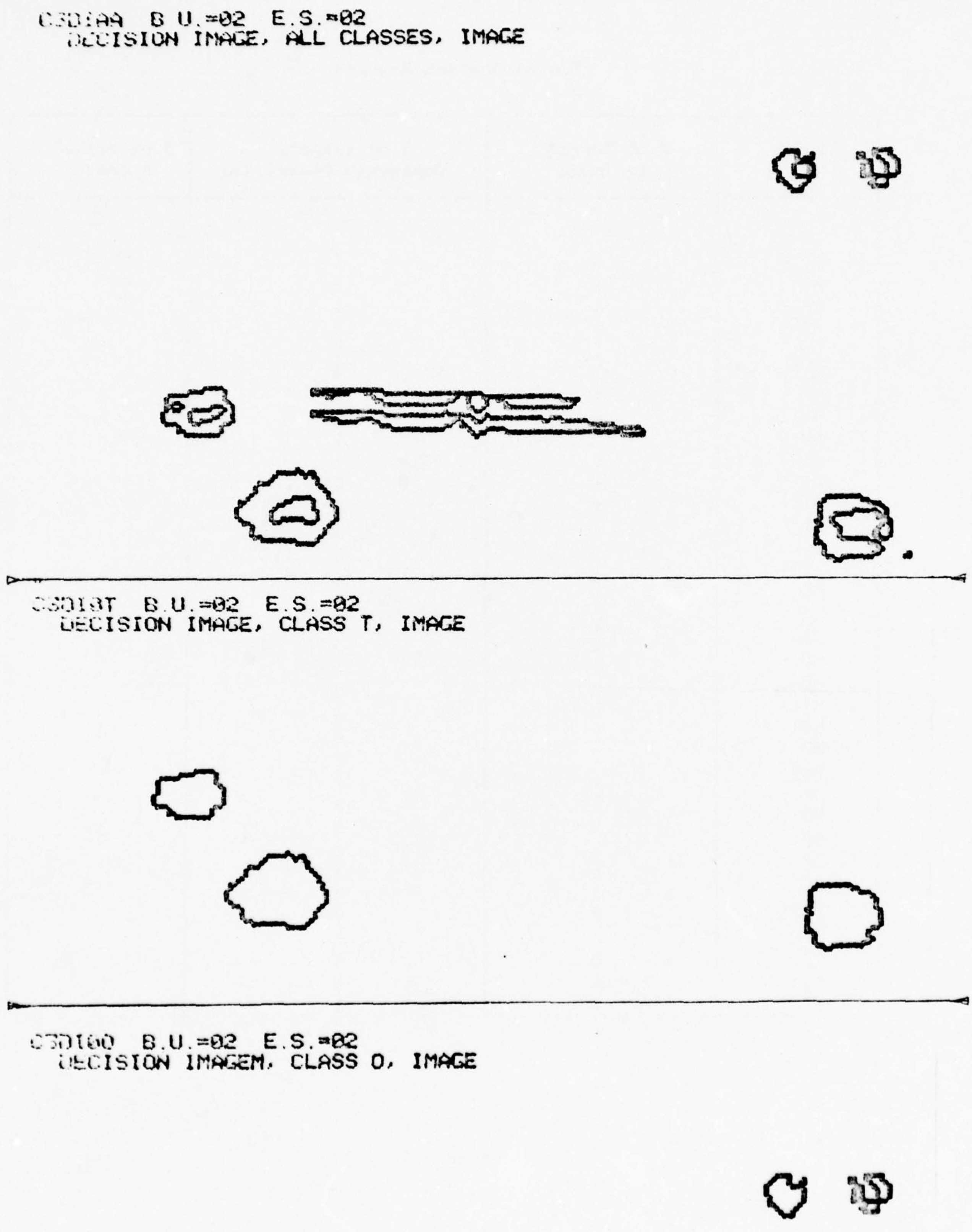


FIGURE 4-38 SOME DECISION IMAGES FOR IMAGE C4; (a) TOP, (b) CENTER, (c) BOTTOM

Classification Results

Image	# of Targets in Image	# of Targets Correctly Classified	# of False Alarms
L7	1	1	
L8	1	1	
L9	1	1	
L10	1	1	
K5	1	1	
K6	1	0	
K7	1	1	
K8	1	1	
K9	1	1	
K10	1	1	
E1	1	1	
E2	2	2	
E3	1	1	
E4	0	0	
E5	1	0	
E6	2	1	
E7	1	0	
E8	2	2	
H5	2	2	
H6	2	2	
H7	1	1	
H8	1	0	
H9	2	2	
H10	2	2	
B4	1	1	
B5	1	1	1
B6	1	1	
B7	1	1	
B8	1	1	
C1	2	2	
C2	1	1	
C4	3	3	
D7	1	1	
D8	1	1	

Table 4-6

The single false alarm was caused by the marker combining with background noise to form a target-like boundary. This false alarm occurred in image frame B5 and is shown in Figure 4-44.

From the examples presented here, it appears that the center marker was an artificial yet important source of many problems; however, the routine which appears to be the weakest link in the total process was object definition, as performed by "Area Edge Detection". That is, in each case, the miss or false alarm was caused by an improper boundary definition for the object.

Figure 4-39 Missed detection in Frame K6.
Top shows all extracted boundaries. Bottom shows all boundaries classified as 0 overlayed on image.

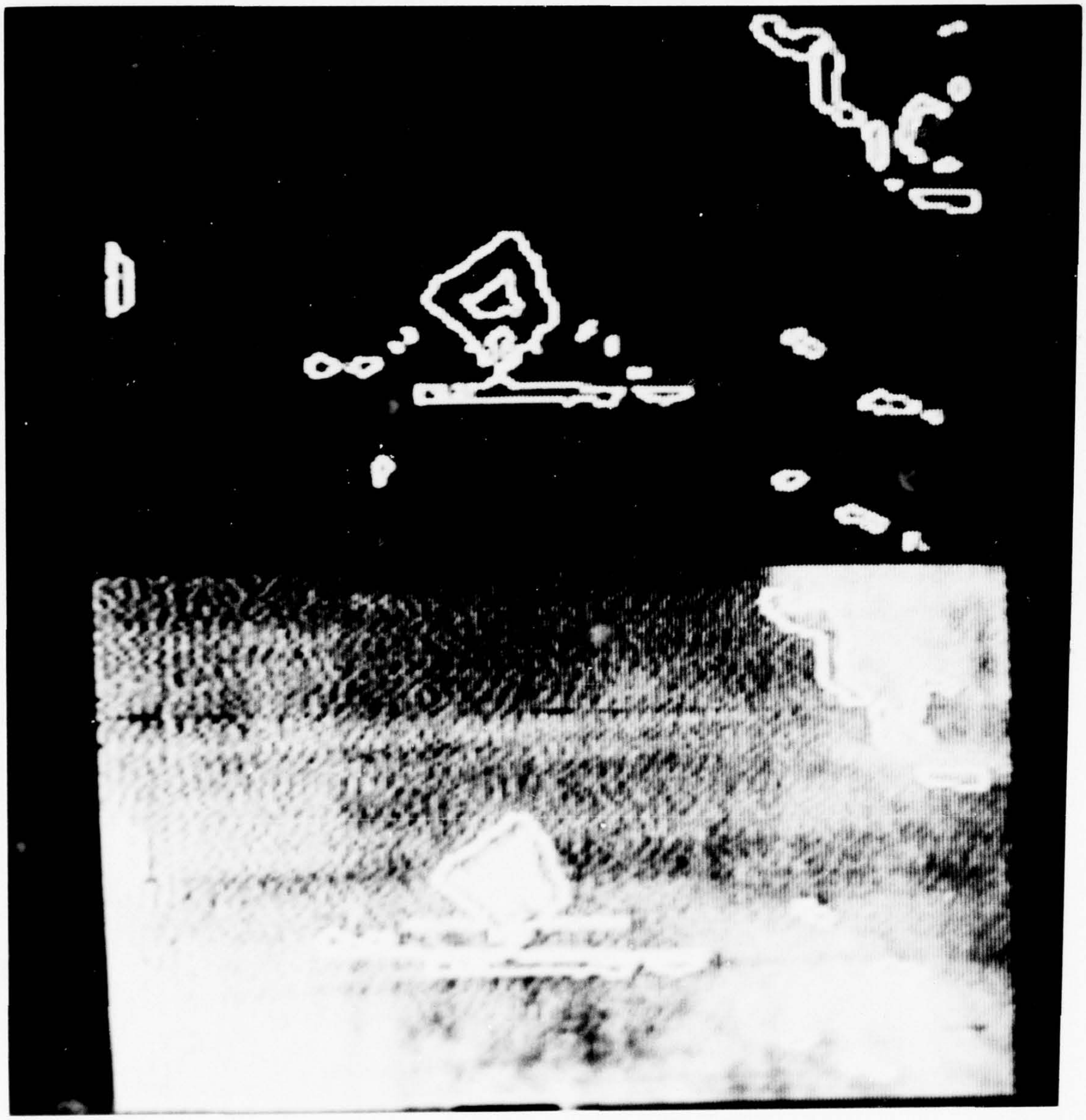


Figure 4-40 Missed detection in Frame H8.
Top shows all extracted boundaries. Bottom shows all boundaries classified as 0 overlayed on image.

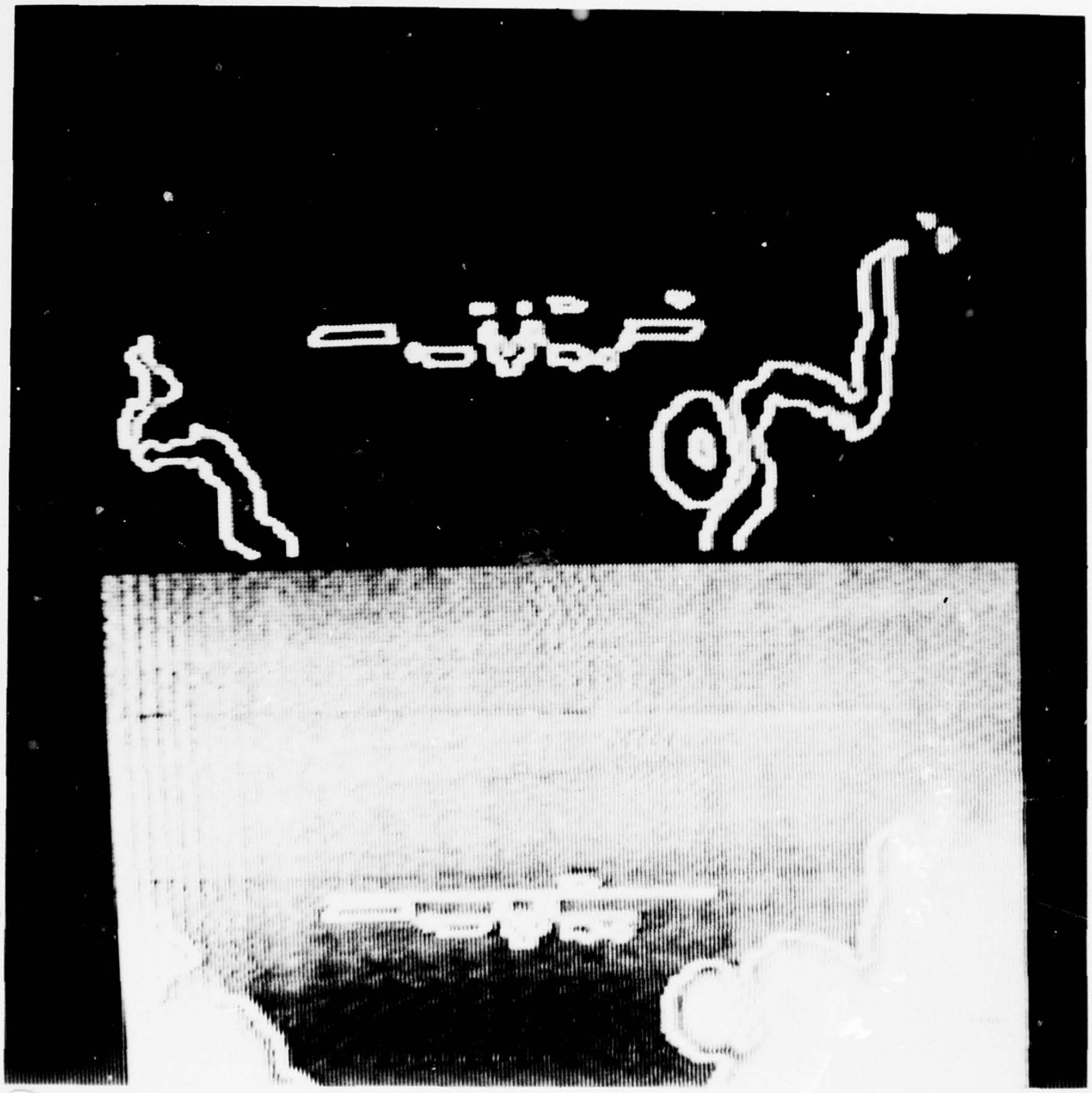


Figure 4-41 Missed detection in Frame E7.
Top shows all extracted boundaries. Bottom shows all boundaries classified as 0 overlayed on image.

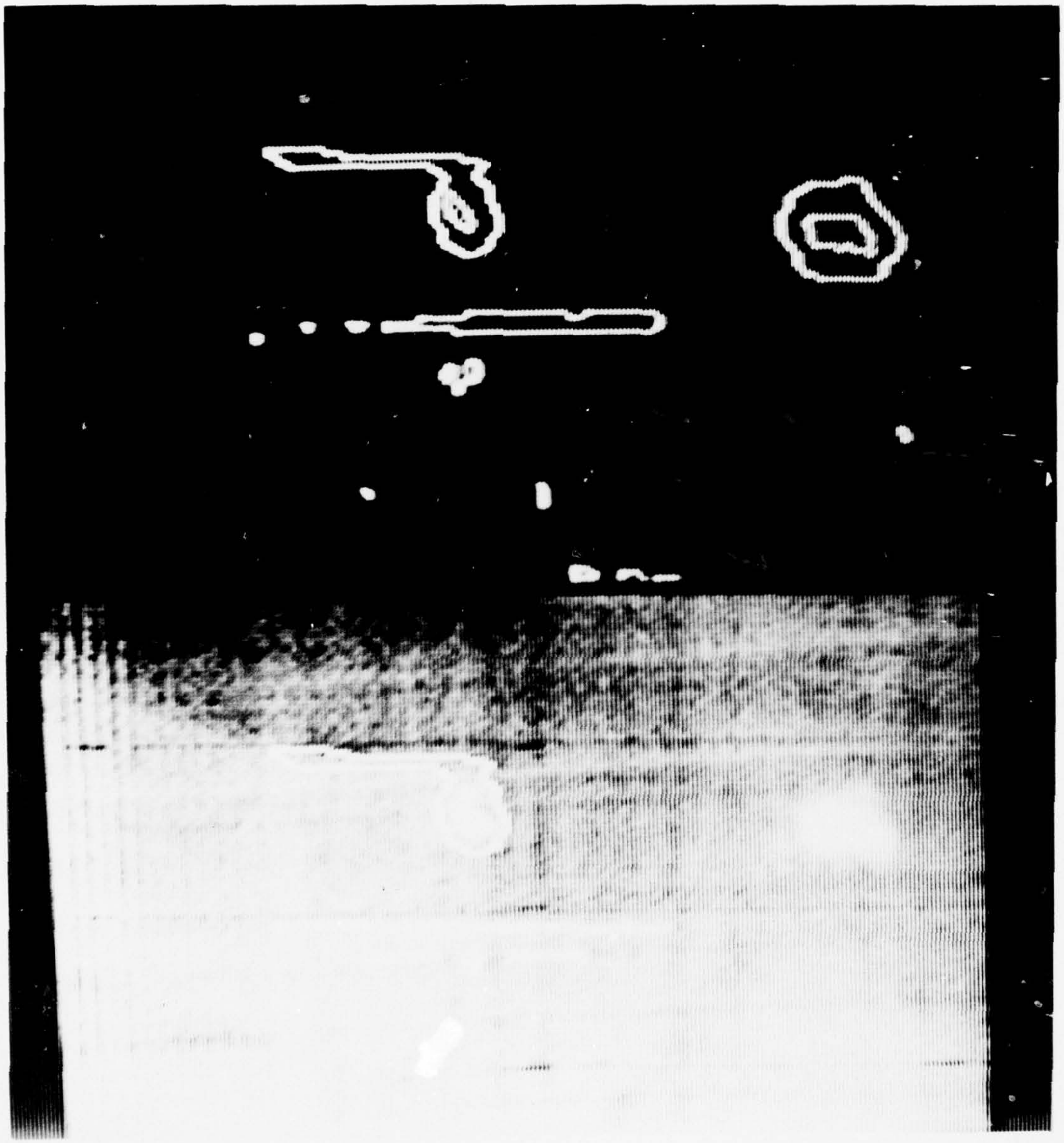


Figure 4-42 Missed detection in Frame E6.
Top shows all extracted boundaries. Bottom shows all boundaries classified as 0 overlayed on image.

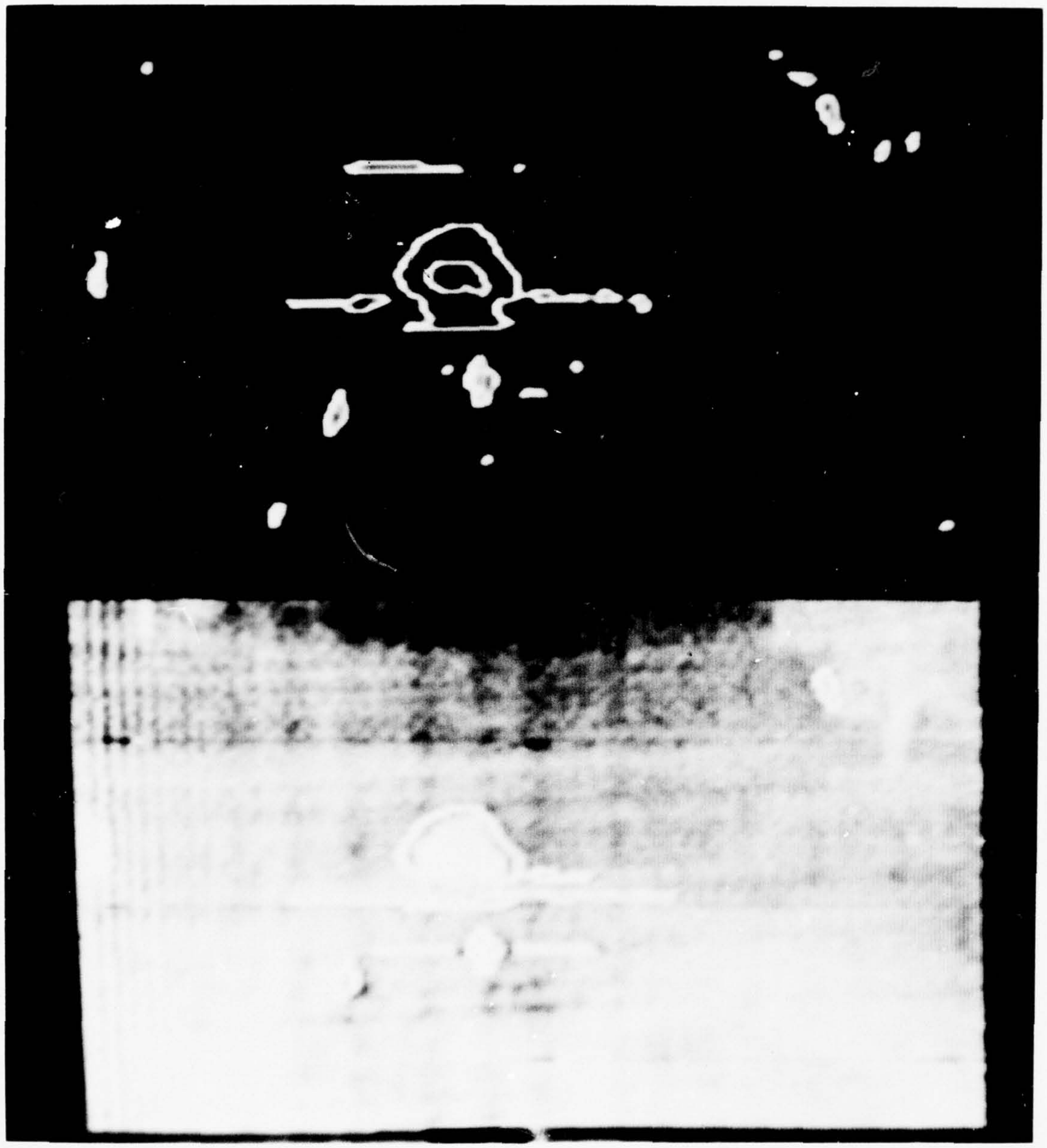


Figure 4-43 Missed detection in Frame E5.
Top shows all extracted boundaries. Bottom shows all boundaries classified as 0 overlayed on image.

Figure 4-44 False alarm (1) in Frame B5.
Top shows all extracted boundaries. Bottom shows
all boundaries classified as targets overlayed on image.

SECTION 5

FURTHER CONSIDERATIONS FOR TARGET DETECTION

In addition to the processing sequence described in Section 4, further considerations were made regarding algorithms which might augment that sequence. In particular, prescreening techniques which might be implemented prior to the edge detection processing were considered. Descriptions of these and alternate techniques and discussions concerning their implementation are presented in this section.

5.1. PRESCREENING

One important consideration of any proposed software target cueing scheme for a FLIR system is the execution time. To be practical, any scheme must be able to operate at real-time or near real-time rates. One method investigated to improve execution time was the use of certain simple statistics to identify subareas of an image which could contain targets. Those subareas which were identified as not containing potential targets would be screened out while the rest of the possible target areas would be passed on for further processing.

This type of prescreening was investigated using the "Search" routine on the DICIFER system. This routine passes a rectangular window across an image (or sequence of images) at specified intervals. The size of the interval and window are chosen so that a target fits entirely within at least one subimage. For each window position, specified measurements are calculated using the image data within the window. These measurement values are then compared with their corresponding thresholds. If any of the measurement values satisfy the appropriate threshold, then the corresponding rectangular subimage may contain a target.

The measurements currently available within the Search routine are grey level mean, standard deviation, median, lowest grey value, highest grey level, range, and any user supplied algorithm written in FORTRAN. In order to be accepted the sub-image must be a user-specified distance from previously accepted sub-images.

Figures 5-1 through 5-4 show the density histograms and statistics generated for different types of subareas. By using the standard deviation measurement, prescreening was achieved which eliminated from 40% to 90% of the image without eliminating any targets. This is a simple computation to make and if used should reduce the total computation requirements.

5.2. AUTOMATIC SUBIMAGE ANALYSIS

Target/object boundary analysis techniques were considered as a means of discriminating true targets from nuisance objects. The process discussed could not be implemented within the scope of this effort, but it remains as a topic of future interest and of good potential for target detection. An algorithmic description is provided in the following paragraphs.

The presence of a target within an appropriately sized subfield of the imagery is generally accompanied by the bimodal distribution in the grey level histogram of the subfield. Here, one mode is associated with the grey level distribution of the background and the other mode corresponds to target information. The pixels are separable into target and background by a statistical decision theory thresholding technique. The threshold can be chosen even when there is significant overlap between the two distributions, and the bimodality is not obvious, as in the case of a low-contrast target.

It is assumed that the subfield size can be determined from knowing true ground scale. This can be determined from the altitude and depression angle with respect to the terrain surface in the field of view, and knowing the acceptance angle of the sensor in use.

Such information can be made available to the FLIR from other subsystems. One subfield should completely circumscribe the target plus some background pixels as well. In the immediate vicinity of the target it is helpful to assume that the background grey level distribution is symmetrical about its mode. Furthermore, there will most likely be several times more background pixels than target pixels, allowing the background points to have a distribution with a distinct peak closely corresponding to the mode. Thus, roughly half of the background pixels in the subfield should have intensity values less than the mode and half above. The number of pixels whose value is less than the background peak can be counted from the subfield histogram. By doubling this number and subtracting it from the number of pixels in the total subfield, the number of potential target pixels is calculated. If this number is sufficiently large for targets of interest (say, as a percentage of the total subfield) then subsequent target cueing operations would take place. Otherwise, it would be assumed that there is only one distribution and no target exists in that subfield.

If the potential target pixel count indicates further processing, a statistically optimal threshold is computed by folding the known background distribution about the mode, subtracting from the total, and selecting the intensity value equal to half the total; i.e., when the two separate distributions are equal. This threshold is used to create a binary image around which a boundary is traced. If the boundary

closes upon itself within the subfield, its extent, length, and shape-sensitive features are measured. These are input to the detection logic for a yes-no-maybe decision. A flow diagram illustrating the overall processing scheme is given in Figure 5-5. A detailed account of the sub-image threshold calculation which is critical to this technique is provided in Appendix B.

It should be stressed that although the presence of a detectable target within a sub-field produces a skewed or even bimodal distribution, the existence of a target can not be inferred from the presence of such a distribution. However if there is sufficient symmetry, it is likely that there is no target within the subfield. In other words, the set of subfields containing detectable targets, i.e., unobscured targets with sufficient contrast with their immediate background, is contained within the set of sub-fields exhibiting skid or multi-modal grey-level distributions.

MEAN	STD	MEDIAN	MODE	POP
91	22	85	84	7476

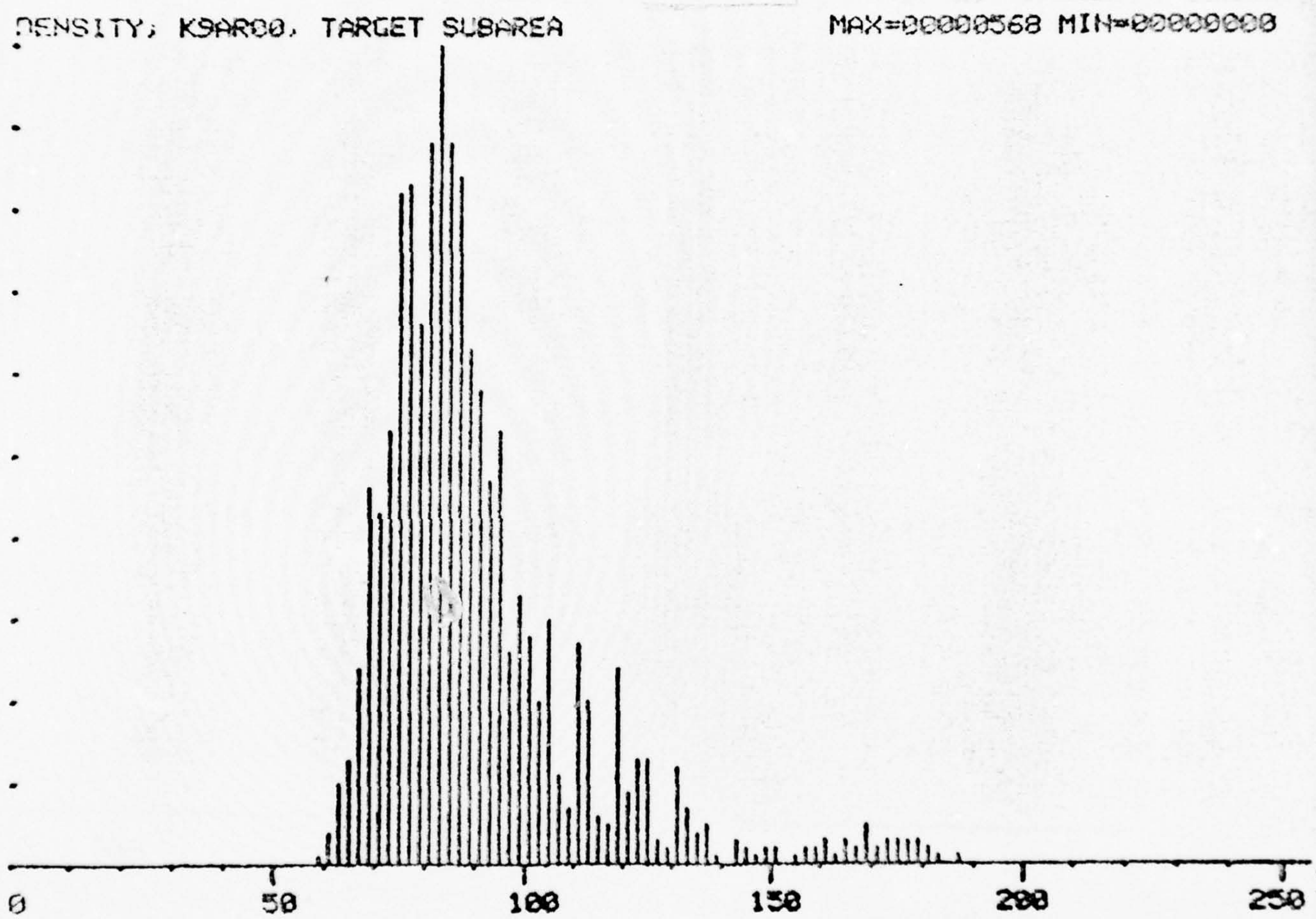


FIGURE 5-1 HISTOGRAM AND STATISTICAL CALCULATIONS FOR A SUBIMAGE CONTAINING A TACTICAL TARGET

MEAN	STD	MEDIAN	MODE	POP
86	8	87	84	4559

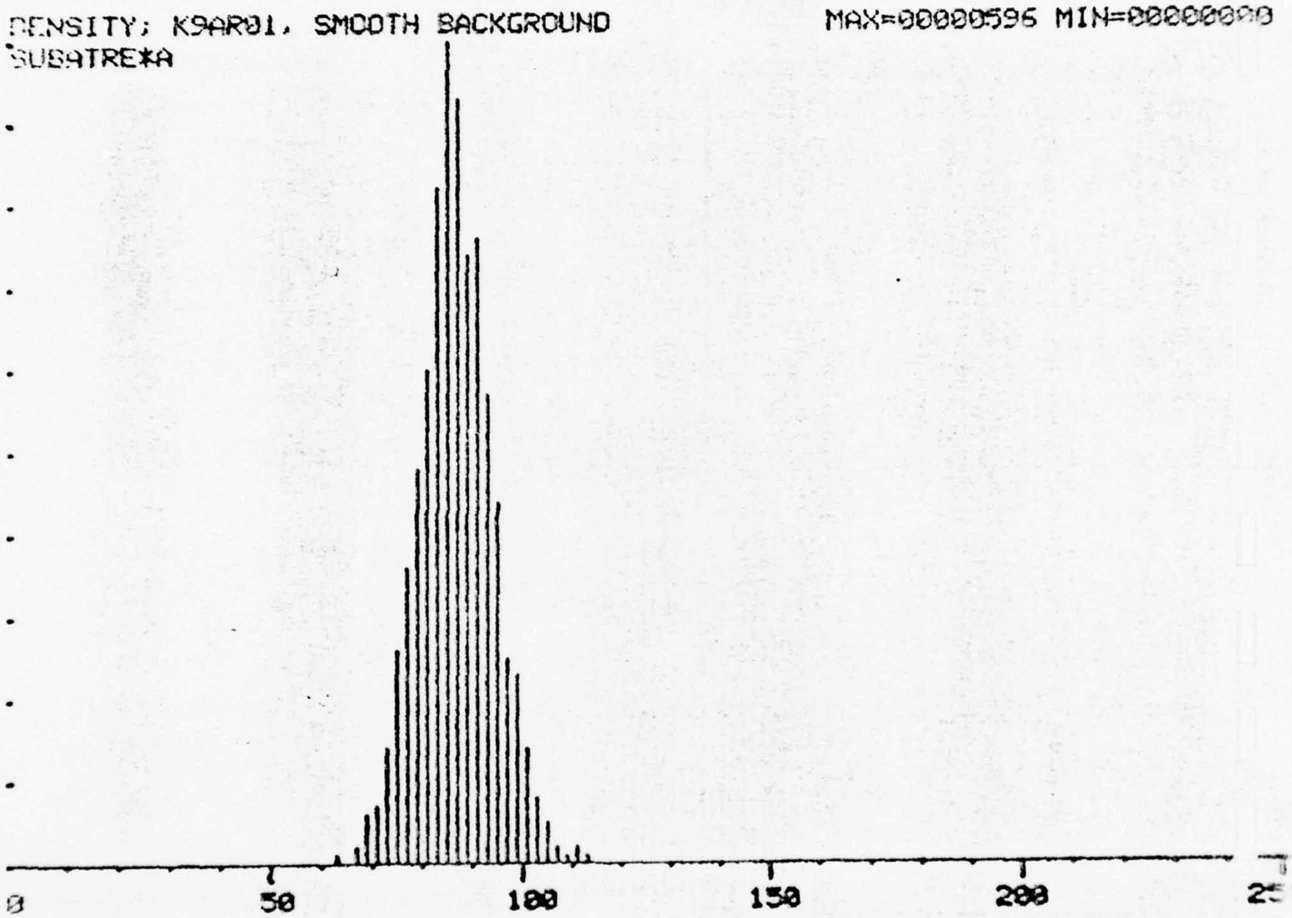


FIGURE 5-2. HISTOGRAM AND STATISTICAL CALCULATIONS FOR A SUBIMAGE CONTAINING ONLY "SMOOTH" BACKGROUND

MEAN	STD	MEDIAN	MODE	POP
93	18	90	87	4200

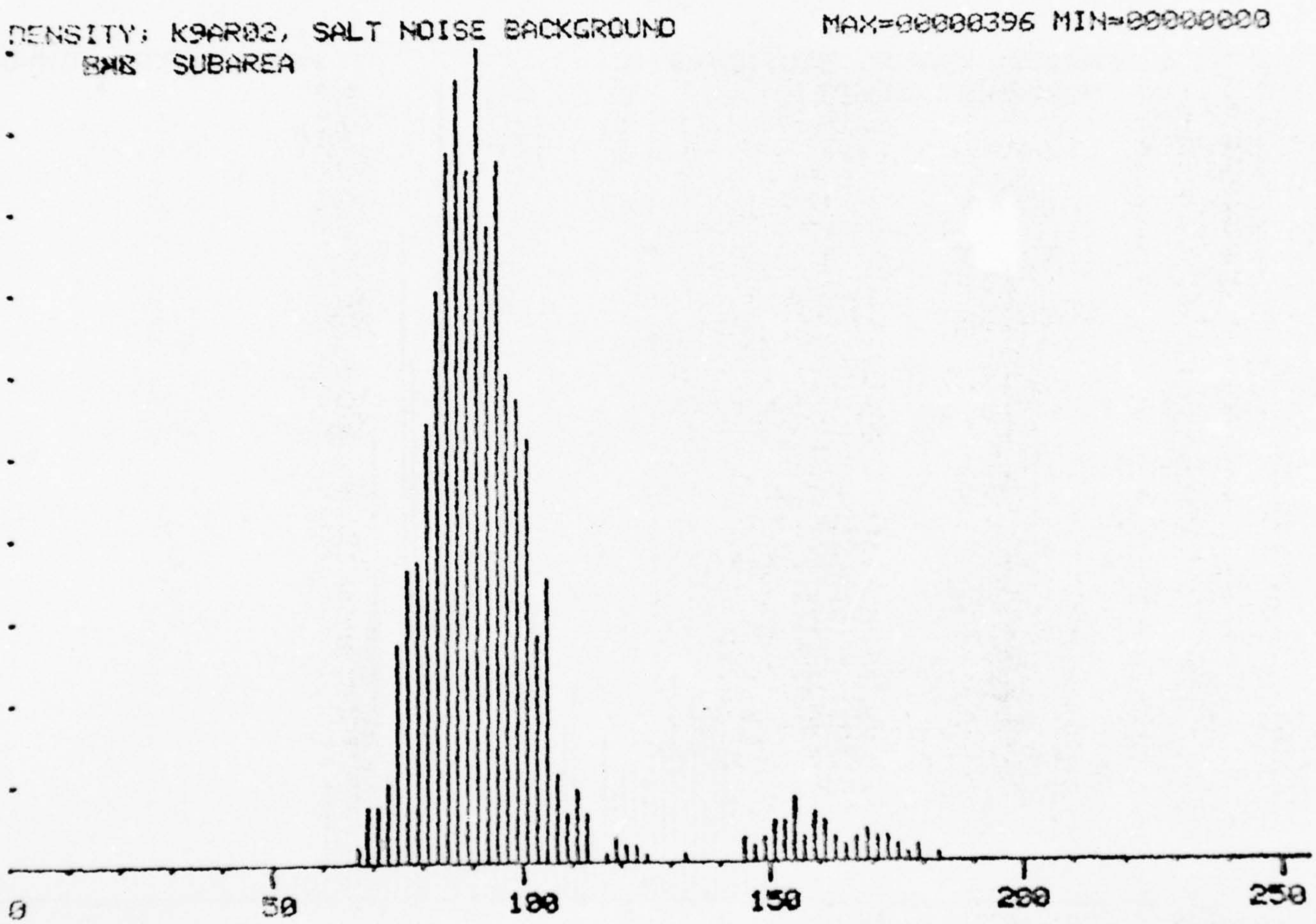


FIGURE 5-3 HISTOGRAM AND STATISTICAL CALCULATIONS FOR A SUBIMAGE CONTAINING A "NOISY" BACKGROUND

MEAN	STD	MEDIAN	MODE	POP
124	24	131	136	4860

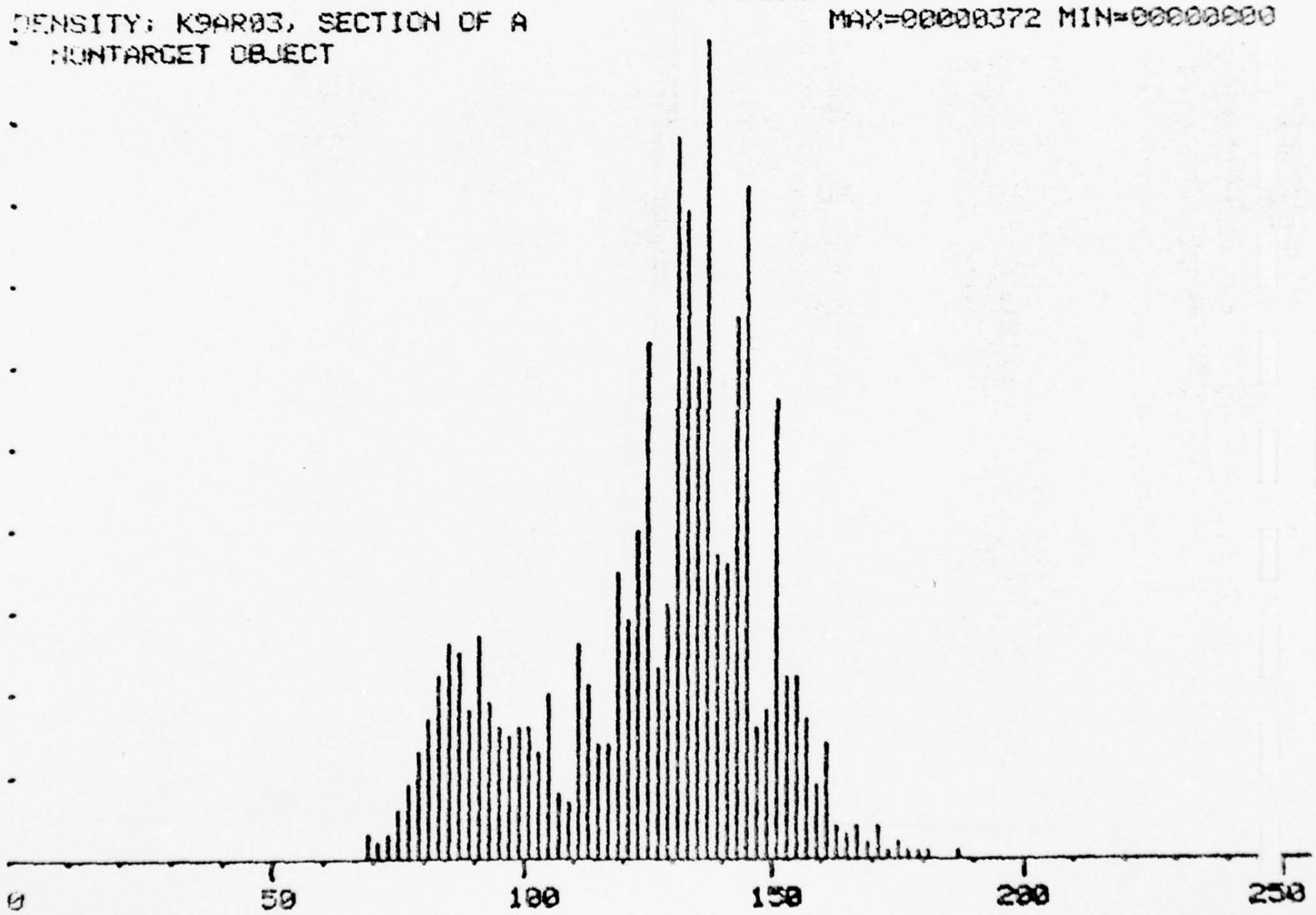
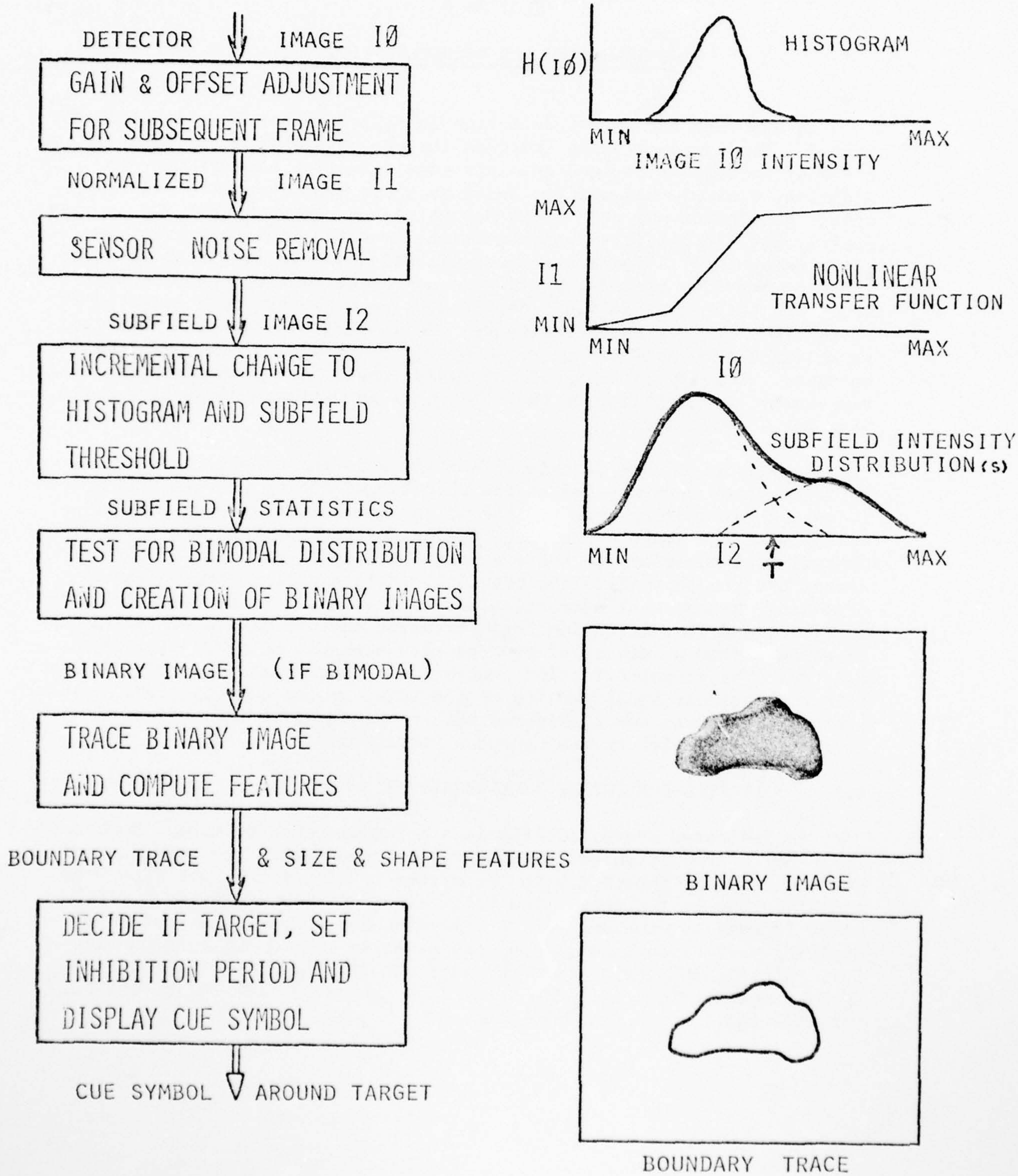


FIGURE 5-4 HISTOGRAM AND STATISTICAL CALCULATIONS FOR A SUBIMAGE CONTAINING A PORTION OF A NON-TARGET OBJECT

Figure 5-5

PROCESSING SEQUENCE PROPOSED FOR TARGET CUEING



SECTION 6

CONCLUSIONS AND RECOMMENDATIONS

An approach for target detection in FLIR data has been defined and tested. The approach taken included the use of simple noise removal, contrast normalization, and gradient edge detection algorithms. In addition, a simple classifier, based on shape features extracted from object boundaries was used. The detection accuracy achieved was approximately 88% (38 out of 43 representative targets) with only one false alarm occurring. A false alarm was allowed anywhere in the total image. Three of the five missed targets were contaminated with graphics that would not be present in the video data. The other two were detected as "potential" targets but subsequently misclassified. Use of a globally established threshold was required by the constraint of the existing software. Locally established thresholds would have given correct responses. Descriptions of the algorithms and illustrations of results have been presented.

During the course of this effort the existing capabilities of the DICIFER system were applied to the FLIR target cueing problem. It is important to remember that DICIFER is an interactive general-purpose system with the image processing, feature extraction, and logic design capabilities necessary to solve a large variety of problems. This system has the capability for multi-file data handling, preprocessing, searching, measurement extraction and evaluation, feature data structure analysis, and classification logic creation and evaluation of digitized image data from a variety of sources with emphasis on remotely sensed imagery. The target detection process described in this report utilized only a relatively small portion of the total system capabilities. Furthermore it was not considered within the scope of this effort to augment DICIFER with special purpose routines.

6.1. TOPICS FOR FUTURE INVESTIGATIONS

As indicated above, DICIFER is a powerful research tool. However, it is not a production-oriented system. For example, the user/analyst must continually interact with the system to direct the data flow from one algorithm to the next. The total flexibility provided in DICIFER makes it easy for the analyst to determine and compare the results of applying various combinations of processes to a data set. In particular, this flexibility allowed for implementing the noise removal, contrast normalization, and edge detection processes which define the FLIR target detection process described.

Once the total processing scheme for a particular application, as here with the FLIR, has been defined, that process may be applied to a set of test data using the DICIFER system. However, since DICIFER is not oriented toward production processing, real time throughput rates cannot be achieved. This fact limited the volume of data which could be efficiently processed during the term of this effort. For this reason, it is recommended that the processes described in this report be implemented with special-purpose parallel processor units to assess the real-time capabilities of the detection scheme. This would allow the processing of a large volume of FLIR data.

6.1.1. Real-time Sequential Processing

As previously described, the total target detection process consists of a cascade of subprocesses. It is desirable that each of the subprocesses be basically a neighborhood operation requiring the storage of relatively few scan lines. The subprocesses used meet these requirements with the exception of the boundary chain encoding algorithm. Nevertheless, investigations by PAR personnel made independent of this contractual effort indicate that an efficient boundary-processing algorithm for grey-level images can be implemented to meet these real-time requirements. It is recommended that future investigations be made to evaluate the feasibility of this approach.

Because size and shape were the attributes used ultimately to separate targets from non-targets, it is plausible to expect some discrimination capability within the target class. Thus we recommend extending this effort to include classification of target type, e.g. tank, truck, APC. The local thresholding technique described in Section 5 would provide better shape definition than the gradient method used for detection only.

AD-A037 136 PATTERN ANALYSIS AND RECOGNITION CORP ROME N Y F/G 17/5
PROCESSING OF FLIR DATA ON DICIFER.(U)
JUL 76 D J LUCEY, G E FORSEN, M J ZORACKI DAAG53-75-C-0277
UNCLASSIFIED PAR-76-26 NL

2 OF 2
ADA037136

END
DATE
FILMED
4-77

REFERENCES

1. Forsen, G.E., J.C. Lietz, et al., "Image Feature Extraction System (IFES)" RADC-TR-74-291 (AD002 878), Sept. 1974.

2. Sammon, J.W., Jr., "On-Line Pattern Analysis and Recognition System (OLPARS), RADC-TR-68-263, Aug. 1968. (AD675 212).

3. Haralick, R.M., "Texture-Tone Study with Applications to Digitized Imagery", Technical Report 182-1, Center for Research, Inc., University of Kansas, Lawrence Kansas, Dec. 1970.

4. Sammon, J.W., Jr., "Interactive Pattern Analysis and Classification", IEEE Transactions on Computers, Vol. C-19, No. 7, pp. 594-616, July 1970.

5. Sammon, J.W., Jr., "A Nonlinear Mapping for Data Structure Analysis", IEEE Transactions on Computers, Vol. C-18, No. 5, pp. 401-409, May 1969.

6. Sammon, J.W., Jr., "An Optimal Discriminant Plane", IEEE Transactions on Computers, Vol. C-19, No. 9, pp. 826-829, Sept. 1970.

7. Zanon, A., M. Gillotte, and M. Zoracki, "Spectral Analysis," RADC-TR-75-302, March 1976.

8. Lietz, J., et al., "Applications of DICIFER", RADC-TR-76-4, Jan. 1976.

APPENDIX A

BOUNDARY TRACING ALGORITHM

Edge Follower

The basic concept of boundary tracing is quite straightforward. The algorithm essentially consists of two parts: a scan mode and a trace mode. The scan mode locates a boundary within an image and the trace mode follows the boundary and generates the chain vector.

The scan mode employs a basic raster scan. Adjacent pixels in a row are compared to one another to detect a change from background to object (or from non-edge to edge). The input image is assumed to be binary. When a difference is detected, the algorithm checks both pixels to see if either pixel has been marked by the tracing mode as a previously-traced boundary pixel. If one has been so marked, the scan simply skips the point and continues scanning, thus avoiding repeated tracing of the same boundary. If neither pixel has been marked, an untraced boundary has been detected, and the scan mode transfers control to the trace mode. The scan mode provides the trace mode with the initial starting pixel (which is always taken to be the object pixel rather than the background pixel) and an initial starting direction. When the tracing of the object is finished, the trace mode returns control to the scan mode, which then continues scanning where it left off.

After a row has been completely scanned, the program continues scanning at the beginning of the next row down. This raster scanning continues until the entire image has been processed.

Detailed Description of the Trace Mode

The trace mode consists of finding which of the eight pixels adjacent to the present boundary point is the next boundary point and recording this in a chain vector file. The scan mode provides the initial starting point, $I_o = (x_o y_o)$, and the initial direction link to the trace mode. The direction links are assigned values in the following manner:

x

y

1	2	3
0	(x,y)	4
7	6	5

Clearly, incrementing the direction links (with an appropriate jump for the 7 to 0 transition) causes a clockwise rotation around (x,y). The trace mode begins by locating the pixel (A,B), associated with the initial direction

link. If the pixel (A,B) is a background point, then the direction link is incremented and the new point (A,B) associated with the new trial link is examined. If none of the eight adjacent pixels are object points, then point (x,y) is classified as an odd-dot point; as such, it is removed from further consideration and control returns to the scan mode.

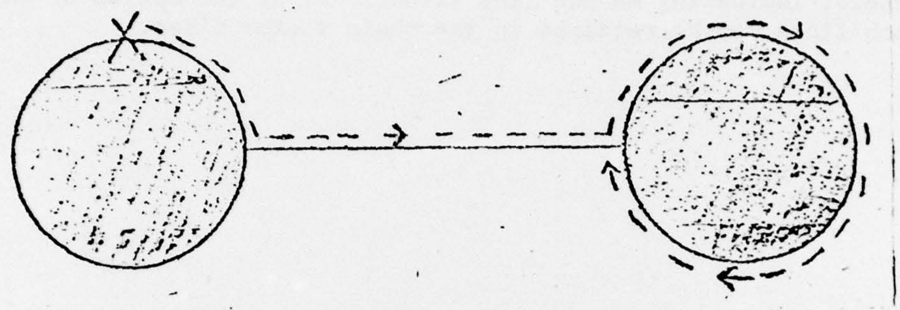
When point (A,B) represents an object point, the value of the link is recorded in the associated chain vector file. Then pixel (x,y) is marked as having been traced and pixel (A,B) becomes a new center pixel (x,y). A new initial link direction is also calculated. At this time the algorithm returns to the point (x,y), where it searches the eight adjacent pixels to find the next boundary pixel. This operation continues until the point (A,B) found by the search is point I_o. Thus, the boundary of an object is traced in a clockwise sense until it returns to its starting point.

Special Considerations

There are certain special circumstances which the basic boundary tracing algorithm must identify. The problem of handling a boundary which runs into the edge of the image is the first such circumstance we will consider. In this case, the contour cannot be followed completely around to the initial starting point. One possible solution is to follow the edge of the image until another point of intersection between the object and the edge of the image is found, thereby allowing the trace to continue. In this manner the edge of the image becomes part of the object's boundary. It was felt that the shape described by such a contour would be more misleading than helpful. Instead, such "edge-limited" boundaries would be described by a special chain vector which would have two end-points, each at the edge of the image. This scheme would not include the image edge as part of the contour.

The most efficient way to detect edge limited boundaries is to use an image edge scan. This algorithm initially scans the edges of the image to locate all the edge-limited boundaries before raster scanning occurs. As each boundary is found, it is traced to its second point of intersection with the image edge. If this were not done, a normal raster scan could start tracking an edge-limited boundary somewhere between its endpoints. Then when the image edge was detected, the chain would have to be reversed to maintain the proper order, and then a counter-clockwise search about the boundary starting with the initial starting point would have to be performed to obtain the remainder of the boundary. With an edge-search, these problems are eliminated, since the chain always begins with one of the endpoints and the boundaries so traced would not be detected again. The edge scan begins by scanning the first row and comparing adjacent columns. Then it scans the first and last column, comparing adjacent rows. Finally, it scans the last row, comparing adjacent columns.

The second special condition concerns retracing pixels. As mentioned, to prevent repeated traces of the same boundary, the pixels are marked by the trace mode as they are acquired. However, under certain circumstances, the trace mode must consider such marked pixels as valid object points which should be traced again. This situation is most easily illustrated by example. Consider an object with a dumbbell shape as shown below:



The trace would proceed along the boundary as shown by the arrows until it returned to the connecting bar. If this bar is only one pixel thick, then the only way to complete the boundary would be to retrace the pixels along the bar. As an indication that this has occurred, all retraced links in the chain vector are marked for future reference.

Another special consideration is the case of an object with an inside boundary, such as a doughnut shape. The basic algorithm will work on this inside boundary, but the direction of the trace around the boundary will be counter-clockwise. This counter-clockwise characteristic of the chain provides a means of differentiating between inside and outside boundaries.

The reason inside edges are traced in the counterclockwise direction may best be explained as follows. For a given object the first boundary point encountered during the scanning process (top to bottom, left to right) forms the initial point of the boundary. This point would, of course, be on the outside of the object. At that time the entire outside boundary is followed and the object boundary points marked as having been traced. Having completed the encoding of the outside boundary for that object, the algorithm returns to the scanning mode. No other outside boundary can now be generated for the object just created. If, however, there are "holes" in that object, there will be edge (inside) points which have not been traced. Once the first such inside edge point is found, the trace mode takes over. The fact that we are on an interior "surface" forces the trace to proceed along the boundary in a counterclockwise direction since the trace proceeds in a direction such that non-edge points are always to the left side of the boundary.

A final special consideration is how to handle an object or part of an object which is only one pixel thick. Such a line, which does not connect two objects, will be considered as an odd line. In many cases, such odd lines are simply noise and one would like to eliminate them. It is possible to provide an odd line elimination routine as part of the boundary tracing algorithm. If the directions of two successive links in a chain are 180° apart, then the boundary has just doubled back on itself, indicating an odd line situation. At the option of the user, such lines may be retained in the chain vector files.

APPENDIX B

SUB-IMAGE SINGLE THRESHOLD CALCULATION

Purpose

To calculate the optimum threshold for creating two-valued (binary) images of targets having higher intensity than their surrounding backgrounds. This routine would be used only when target images are large enough to have size and shape, and when there are no shadows or undershoot.

Assumptions

1. A single threshold value is not usable over the total image.

2. The histogram for the sub-image has been previously calculated.

3. The presence of targets is basically detected by locating areas of increased intensity relative to their immediate surrounds, i.e., noting where "detectable" local contrast exists due to sufficient target temperature. If so, a previous sub-image masking test used for pre-screening would have given a positive indication.

4. When a target is totally contained within a given sub-image, its intensity histogram will be bimodal, flat-topped, highly skewed, or with a distinct "tail" in the higher intensities.

5. The area of higher temperature creating the target "image" is a significant but not too large fraction, e.g., about one-fifth to one-third, of the total number of points in the sub-image.

6. The most restrictive assumption for this function is that the surrounding background intensity distribution is symmetrical about its mode and that its mode is equivalent to its mean. This means that the returns from the immediate neighborhood are uniform except for random variations. Previous use of a uniform background test would have given a positive indication.

7. It is equally costly to label a target element as belonging to the background class as it is to label a background pixel as belonging to the target class within any one sub-image. This is because target signature area is a desired feature to be determined and the decision threshold should not be biased to give the wrong size.

Comments

The use of this routine to calculate a single threshold for a given sub-image requires positive indications from several previously performed tests. First, the masking test would have indicated a bright area of the right size surrounded by a darker area corresponding to background. Second, the uniform background test, which could require calculating the histogram for the points on the sub-image periphery and testing the hypothesis that they belong to one distribution, would have indicated that the area surrounding the central area is sufficiently uniform. This test helps ensure that subsequent processing is performed only when the target signature is centered in the sub-image.

Having passed these two tests there is a good chance that the bright "blob" in the middle of this sub-image is an object of interest and worth investigating in more detail. The remaining processing includes further refinement in the detection process using several possible feature extraction techniques. Those features derived from binary images are often dependent on the threshold value used. Thus it should be calculated precisely. This detection threshold is not a target detection threshold but rather a pixel assignment rule threshold. Only one target is allowed per sub-image, though there are many pixels per sub-image.

Method

The essence of this method is to decompose a single histogram for a given sub-image into two separate histograms, one for the background, the other for the target, and then determining the threshold by using the pixel count of target signature area.

The first step in the decomposition process is to find the histogram mode, i.e., the most frequently occurring intensity value. Assumption 5 assures that the modal value belongs to the background histogram. If this assumption cannot always be made, a more complicated algorithm than described herein is necessary which essentially postpones the decision of which histogram has the modal value until after the decomposition is made. Correct calculation of the sub-image size and a narrow range of target sizes for each sub-image size used will help ensure that assumption 5 holds.

The next step in the decomposition process makes use of assumptions 3, 4, and 6. The words "sufficient target temperature" in assumption 3 are interpreted specifically to mean that relatively few (if any) target signature pixels have intensity values less than that of the mean value of the local background. The total histogram between zero and the mean of the background intensities would be (nearly) identical to the background histogram. It also allows the symmetry assumption 6 to be

applied directly. If half is known, the upper half is obtained simply by "folding" the known half about the modal value. This completes the decomposition process assuming what is left of the total histogram are target signature elements. See Figure B-1 for an example.

The next step in the process is to count the number of points remaining after the background histogram is subtracted from the total. This number is a measure of the target signature area. If this area is too large (or too small) to be a valid target signature, then the sub-image is dropped from further consideration.

This estimate of target signature area does not include any shape or intensity distribution criteria. The lower threshold should only be used to eliminate those potential false alarms that are much too small to be target signatures such as extraneous point noise.

With overlapping histograms, some pixel classification errors will be made using a single threshold decision node. There are two criteria for choosing a single threshold. First, one can minimize the total number of pixel labelling errors. Second, one can try to preserve the target signature area. These produce the same threshold value only when the number of background pixels erroneously labelled target signature equals the number of target signature pixels erroneously called background when the single threshold decision rule is applied.

If the total histogram $h(x)$ is decomposed correctly into its two components $h(x|B)$, the background histogram and $h(x|M)$, the target signature histogram, the minimum number of total errors occurs for a threshold value T, where

$$h(T|B) = h(T|M)$$

This assumes that for $x > T$, $h(x|M) > h(x|B)$. If not, more than one threshold is required. This is unlikely for signatures generally brighter than the background.

The second way of determining T is to use the total target signature area count, $A_M = \sum_{x=0}^{L} h(x|M)$, in the following way:

$$\text{Choose T:} \quad \sum_{x=T}^{L} h(x) = A_M = \sum_{x=0}^{L} [h(x) - h(x|B)] = N-2 \sum_{x=0}^{K} h(x)$$

where L = maximum value of x, K = the modal value, and N = the number of pixels.

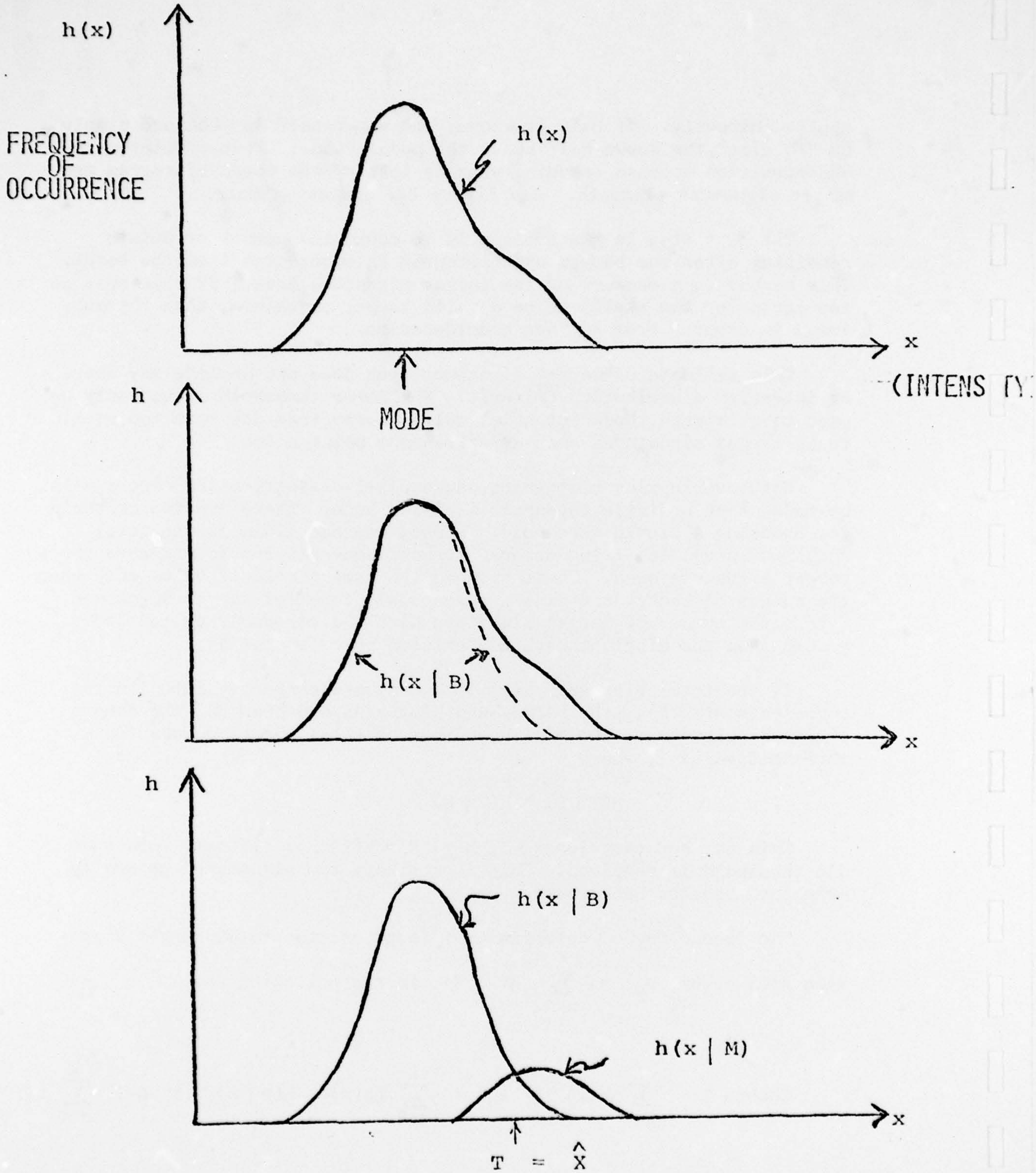


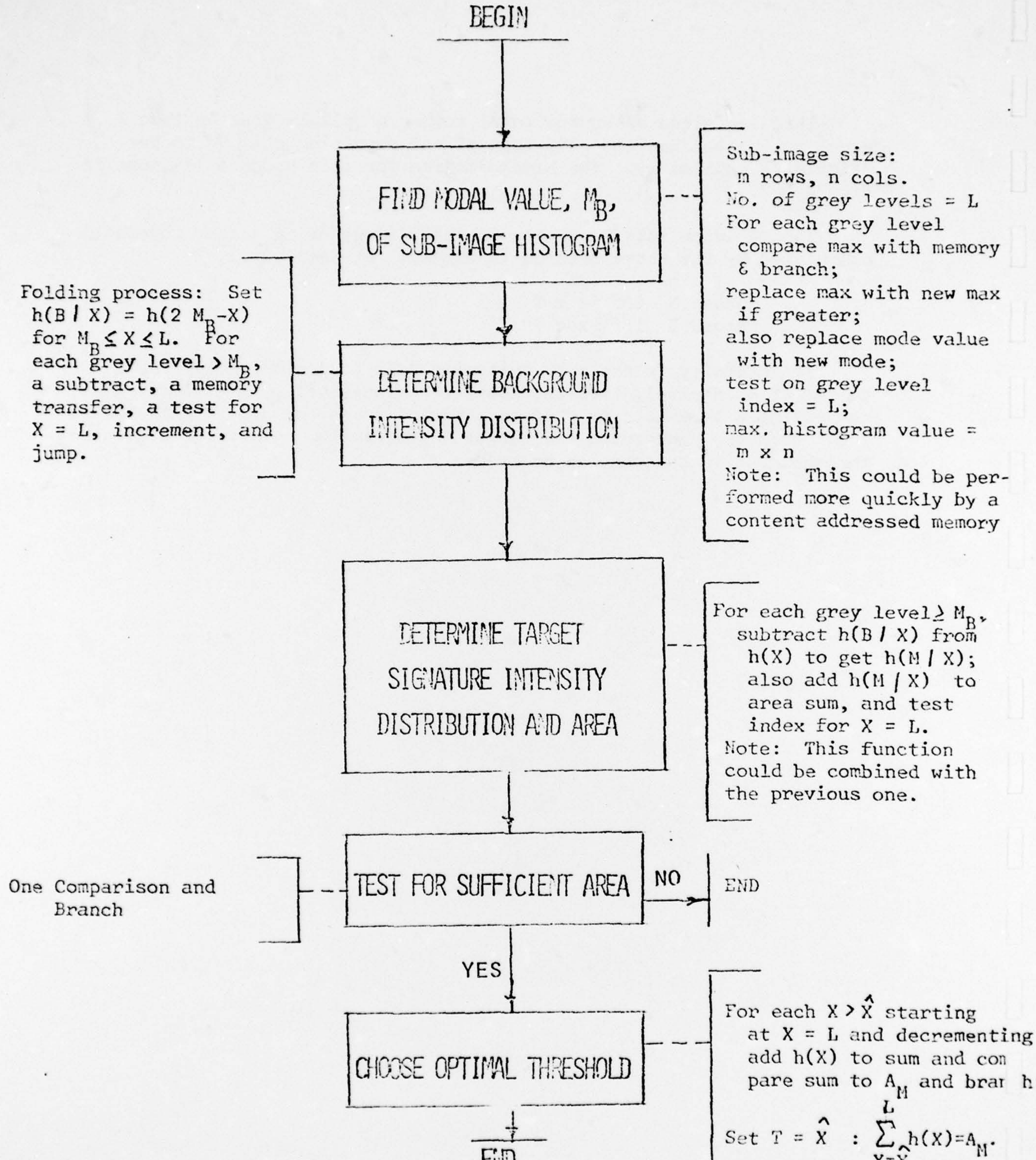
Figure B-1 Decomposition of Bimodal Histogram

This technique makes the total number of pixels greater than T equal to A_M, and has the advantage of not requiring $h(x \mid M)$ to be calculated explicitly. The block diagram for this process is shown in Figure B-2.

The decision rule to create a binary image using either threshold calculated by the above methods is simply: At each pixel,

Choose M iff $x \geq T$
Choose B iff $x < T$

If intensity is the only criterion used to decide background or target at each pixel, then the above techniques offer an effective way to provide a threshold to produce a binary image for subsequent processing. When the background and target histograms do not overlap, simpler thresholding techniques can be used.



Flow Diagram of Sub-Image Single Threshold Calculation

Figure B-2

APPENDIX C

FLIR IMAGE ASSESSMENT

During the initial months of this effort a quality assessment of the FLIR data was made by personnel at PAR. The images were viewed by both inexperienced and experienced interpreters. In one experiment seven viewers scanned transparencies using light tables in an attempt to detect tactical targets. A detection accuracy of approximately 78% was achieved. A general comment arose regarding the excessively high contrast interference noise patterns occurring in much of the data.

In a second experiment two experienced photointerpreters and three members of the PAR research team made further judgments regarding image quality and types of noise present in the FLIR data. This study included an assessment of image contrast, target edge sharpness, and target shape and size definition. Knowledge gained here led to the definition of the noise types described in Section 4 and provided valuable insights to the data structure during the image processing and logic design phases of this effort.